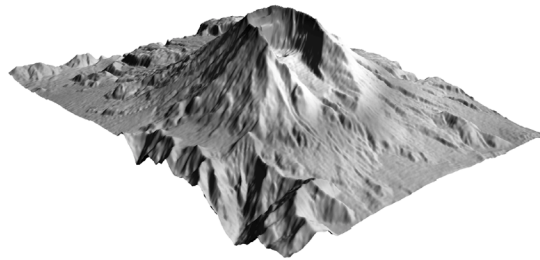


Surfer[®]

User's Guide

Contouring and 3D
Surface Mapping for
Scientists and Engineers



Golden Software, Inc.

809 14th Street, Golden, Colorado 80401-1866, U.S.A.

Phone: 303-279-1021 Fax: 303-279-0909

www.goldensoftware.com

COPYRIGHT NOTICE

Copyright Golden Software, Inc. 1993-2002

The **Surfer**® program is furnished under a license agreement. The **Surfer**® software and user's guide may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software or user's guide on any medium except as specifically allowed in the license agreement. Contents subject to change without notice.

Windows, Visual Basic, Visual Basic for Applications, Excel, and Word are registered trademarks of the Microsoft Corporation.

February 2002

Table of Contents

Chapter 1 Introducing Surfer	1
Introduction to Surfer	1
New Features	2
Screen Layout and Window Types	4
Opening Windows	4
Object Manager	4
Toolbars	5
Scripter.....	5
Using Surfer.....	6
File Types	6
Gridding.....	7
Grid Utilities	7
Map Types	7
Reports.....	11
System Requirements	11
A Note about the User's Guide and Online Help.....	12
Installation Directions.....	12
Network Installation	13
Troubleshooting Installations	14
Updating Surfer	14
Uninstalling Surfer.....	14
Getting Help	15
Online Help.....	15
Context-Sensitive Help	15
Internet.....	15
Technical Support.....	15

Table of Contents

Chapter 2 Tutorial	17
Tutorial Introduction.....	17
Tutorial Lessons.....	17
Understanding How to Use Surfer.....	18
Starting Surfer.....	18
Lesson 1 - Creating an XYZ Data File	19
Opening an Existing Data File.....	20
Creating a New Data File.....	21
Saving the Data File.....	22
Lesson 2 - Creating a Grid File.....	23
Lesson 3 - Creating a Contour Map.....	25
Opening the Map Properties	25
Changing Contour Levels	25
Changing Contour Line Properties	28
Adding Color Fill between Contour Lines.....	30
Add, Delete, and Move Contour Labels	32
Modifying an Axis	33
Saving a Map	36
Exporting 3D Contours.....	37
Lesson 4 - Creating a Wireframe.....	37
Creating a New Wireframe.....	37
Changing the Orientation of a Wireframe.....	38
Changing the Scaling of a Wireframe.....	39
Adding Color Zones to a Wireframe	40
Lesson 5 - Posting Data Points and Working with Overlays	42
Creating Post Maps.....	42
Changing the Post Map Properties.....	42
Overlaying Maps	44
Selecting an Overlay and Assigning an Object ID.....	45
Adding Labels to the Post Map.....	46

Table of Contents

Lesson 6 - Introducing Surfaces	47
Creating a Surface.....	47
Adding an Overlay.....	47
Adding a Mesh.....	49
Changing Colors	50
Chapter 3 Data Files and the Worksheet.....	53
Introduction to Surfer Data and Worksheet Windows	53
Data Files	53
Worksheet.....	56
Working with Data	58
Entering Data into a Cell	58
Moving the Active Cell.....	58
Pasting Data.....	60
Opening Data Files	61
Data File Formats	61
Editing the Data Files in the Worksheet	62
Tips on Selecting Cells	62
Editing the Worksheet	63
Modifying the Worksheet and Data Files	64
Formatting the Data	67
Saving Data Files.....	81
Use Caution when Saving Excel Files!.....	81
File Names, Formats, and File Extensions.....	81
Saving ASCII Data Files.....	82
Page Setup	83
Page	83
Margins.....	84
Options.....	85

Table of Contents

Creating a Data File with Digitize	86
Worksheet Technical Specifications	88
Worksheet Size and Memory Requirements	88
Numeric Limits	88
Chapter 4 Creating Grid Files	89
Introduction to Grid Files	89
A Gridding Example	90
Creating a Grid File	91
Data	92
Filter Data	92
Statistics	95
View Data	95
Grid Line Geometry	95
Grid Limits	97
Grid Density	97
Cross Validate	100
The Cross Validation Process	100
Using Cross Validation	101
The Cross Validation Dialog	101
Statistics Note	102
Cross Validation References	103
Grid Report	103
General Gridding Options	103
Search	103
Anisotropy	108
Breaklines and Faults	111
Gridding Methods	114
Inverse Distance to a Power	114
Kriging	117
Minimum Curvature	121
Modified Shepard's Method	125

Table of Contents

Natural Neighbor	127
Nearest Neighbor	129
Polynomial Regression	131
Radial Basis Function	132
Triangulation with Linear Interpolation.....	135
Moving Average	137
Data Metrics	139
Local Polynomial.....	147
Choosing Gridding Methods.....	150
General Gridding Recommendations.....	150
Choosing Methods Based on the Number of XYZ Data Points.....	155
Exact and Smoothing Interpolators.....	156
Weighted Averaging.....	157
Producing a Grid File from a Regular Array of XYZ Data	158
Gridding a Function	159
Grid Files	159
Grid Files	160
GRD.....	160
USGS DEM	160
GTopo30.....	160
SDTS	160
DTED.....	161
Chapter 5 Variograms.....	163
Introduction to Variograms.....	163
Variogram Components.....	164
Kriging and Variograms	164
The Variogram Grid	165
The Variogram.....	168
Creating a Variogram	171

Table of Contents

Editing Variogram Properties	171
New Variogram Properties	172
Data.....	172
General.....	173
Variogram Properties.....	175
Experimental.....	175
Model.....	180
AutoFit.....	188
Statistics.....	191
Plot.....	192
Default Linear Variogram.....	193
Exporting a Variogram	194
Using Variogram Results in Kriging	194
Suggested Reading.....	195
 Chapter 6 Contour Maps.....	197
Introduction to Contour Maps.....	197
Drawing Contours.....	198
Contour Map Components.....	199
Creating a New Contour Map.....	200
Editing Contour Map Properties	200
Contour Map Properties.....	201
General.....	201
Levels.....	203
Contour Lines	207
Assigning Gradational Line Properties.....	207
Setting Line Properties on a Frequency Basis	209
Assigning Line Properties to Specific Contour Levels	210
Index Contours.....	211
Color Fill.....	212
Assigning Color Fill Based on a Fill Spectrum	213
Assigning Color Fill to Specific Contour Levels.....	214

Table of Contents

Setting Fill Properties on a Frequency Basis	214
Creating a Filled Contour Map Containing Blanked Areas	216
Displaying a Filled Contour Map without Contour Lines	217
Color Scale Bar	218
Adding a Color Scale Bar	218
Editing a Color Scale Bar	218
Color Scale Bar Visibility	219
Removing a Color Scale Bar	219
Labels	220
Displaying Contour Labels at a Regular Interval	220
Assigning Labels to Specific Contour Levels	221
Setting Label Properties on a Frequency Basis	221
Label Spacing	223
Removing All Labels	224
Editing Label Positions	224
Hachures	226
Setting Hachures on a Regular Interval	226
Setting Hachures on Specific Levels	227
Setting Hachures on a Frequency Basis	227
Removing All Hachures	228
Smoothing Contours	229
Export Contours	230
Chapter 7 Base Maps	231
Introduction to Base Maps	231
Base Map File Types	232
Creating a Base Map	232
Editing Base Map Properties	233
Base Map Properties	233
Base Map	234
Changing Line, Fill, Font, or Symbol Properties in a Base Map	235
Assigning Coordinates to a Bitmap File	236

Table of Contents

Placing Boundaries on Other Maps	237
Coordinates	237
Limits and Scale	237
Masking Portions of a Map.....	238
Digitize	239
Creating a Blanking File with the Digitize Command.....	240
 Chapter 8 Post Maps.....	241
Introduction to Post Maps.....	241
Post Map and Classed Post Map Components.....	242
Data Files Used for Posting	243
XY Coordinates in the Data File.....	243
Z Values in the Data File	244
Data Labels in the Data File	244
Symbol Specifications in the Data File.....	244
Symbol Angle Values from the Data File.....	245
Creating a New Post Map or Classed Post Map	245
Editing Post Map or Classed Post Map Properties	247
Post Map and Classed Post Map Properties.....	247
General - Post Map	247
General - Classed Post Map.....	249
Labels.....	250
Classes	252
Updating the Data File.....	253
Classed Post Map Legends	254
Display Legend.....	254
Edit Legend Properties.....	254
Legend Properties Dialog	254
Scaled Symbols.....	256

Chapter 9 Image Maps	259
Introduction to Image Maps.....	259
USGS Elevation Files	259
Image Map Components.....	260
Creating a New Image Map	260
Editing Image Map Properties	261
Image Map Properties.....	261
General.....	261
Image Map Overlays.....	264
Color Scale Bars	264
Adding a Color Scale Bar	264
Color Scale Bar Visibility	264
Removing a Color Scale Bar	264
Editing a Color Scale Bar	265
 Chapter 10 Shaded Relief Maps	 267
Introduction to Shaded Relief Maps	267
Reflectance	267
Colors.....	267
USGS Elevation Files	268
Shaded Relief Map Components	268
Creating a New Shaded Relief Map.....	269
Editing Shaded Relief Map Properties.....	269
Shaded Relief Map Properties	269
General.....	269
Shaded Relief Map Overlays	272
Reference	272
 Chapter 11 Vector Maps.....	 273
Introduction to Vector Maps.....	273
1-Grid Vector Maps	273
2-Grid Vector Maps	273

Table of Contents

Vector Map Components	274
Two-Grid Vector Map Data	275
Cartesian Data	275
Polar Data	277
Creating a New Vector Map	279
Editing Vector Map Properties	279
Vector Map Properties	280
Data - 1-Grid Vector Map	280
Data - 2-Grid Vector Map	281
Symbol	282
Scaling	284
Color Scale Bars	286
Adding a Color Scale Bar	286
Editing a Color Scale Bar	286
Color Scale Bar Visibility	287
Removing a Color Scale Bar	287
Clipping Symbols on Vector Maps	288
Clipping and Map Limits	288
Vector Length Legend	290
Editing the Vector Length Legend	290
Legend Properties Dialog	290
Chapter 12 Wireframes	293
Introduction to Wireframes	293
Wireframe Limits	293
Wireframe Grid Files	294
Wireframe Components	294
Creating a New Wireframe	295
Editing Wireframe Properties	295
Wireframe Properties	296
General	296
Z Levels	298

Table of Contents

Color Zones	300
Specifying Lines	301
Line Properties.....	302
Line Property Precedence	303
Z Line Levels and Properties	303
Specifying Evenly Spaced Z Levels	304
Adding, Deleting, or Modifying Z Level Values.....	305
Assigning Z Level Properties Based on a Line Spectrum.....	305
Assigning Properties to Individual Z Levels.....	306
Color Zones	306
Specifying Color Zones	307
Assigning Color Zones based on a Line Spectrum.....	308
Assigning Line Properties to Specific Color Zones.....	308
Adding and Deleting Color Zones	309
Color Scale Bar.....	309
Adding a Color Scale Bar	309
Color Scale Bar Visibility.....	309
Removing a Color Scale Bar	309
Editing a Color Scale Bar	310
Wireframe Base	311
Smoothing a Wireframe.....	312
Blanking on a Wireframe.....	313
Wireframe Overlays.....	313
Chapter 13 Surfaces.....	315
Introduction to Surfaces.....	315
Surface Components.....	316
Creating a New Surface	316
Editing Surface Properties	317
Surface Properties.....	317
General.....	317
Mesh	319

Table of Contents

Lighting	321
Overlays.....	323
Surface Base	325
Base Properties	325
Surface Overlays, Bases, and "Stitching"	327
Surface Memory Requirements	327
Surfaces and Zoom Levels.....	327
Surfaces and Printing.....	327
Color Scale Bars	328
Adding a Color Scale Bar	328
Color Scale Bar Visibility.....	328
Removing a Color Scale Bar	328
Editing a Color Scale Bar	329
 Chapter 14 Common Map Properties	331
Introduction to Common Map Properties	331
Opening Map Properties	332
View	332
View and Overlays	332
Field of View	332
Rotation	332
Tilt	333
Projection.....	333
Setting the View	334
Using the Trackball Command	335
Scale.....	337
Setting the Scale	337
Using Scaling to Minimize Distortion on Latitude/Longitude Maps.....	338
Adding a Scale Bar to a Map	340
Limits.....	342
Limits and Wireframes	342
Limits and Axis Scaling.....	343

Limits and Map Scale	343
Limits and Post Maps	343
Limits and Base Maps	343
Limits, Scale, and Overlaying Maps.....	343
Setting Map Limits	344
Background.....	345
Masking with Background.....	346
Chapter 15 Axes	347
Introduction to Axes	347
Selecting an Axis	348
Displaying Axes.....	349
Axis Scaling.....	349
Axis Limits	350
Positioning Axes	351
Axis Titles.....	352
Axis Ticks and Tick Labels	353
Major Ticks.....	354
Major Tick Spacing	355
Minor Ticks	356
Tick Labels	357
Tick Label Format	359
Axis Tick and Line Properties	360
Grid Lines	360
Chapter 16 Adding Color to Maps	363
Introduction to Color Spectrums.....	363
Colors.....	363
Opening the Color Spectrum Dialog.....	364
The Color Spectrum Dialog.....	365
Reusing a Color Spectrum	367

Table of Contents

Using Color Spectrums in Map Series.....	367
Know Your Data.....	368
Creating a Color Spectrum File	369
Using the Color Spectrum File in Other Maps	370
Level Files	370
Contour Maps	370
Wireframes	370
Creating Level Files.....	371
 Chapter 17 Positioning and Overlaying Maps	373
Introduction to Positioning and Overlaying Maps	373
Moving Maps.....	374
Placing Several Maps in the Same Plot Window.....	374
Creating Several Maps in the Same Plot Window	374
Placing Several Completed Maps in the Same Plot Window	375
Aligning Several Maps on the Same Page	375
Overlaying Maps	376
Overlay Limits.....	377
Overlay Order	377
Wireframes and Hidden Lines	378
Adjacent Maps.....	378
Overlaying Maps	379
Editing Overlays	380
Hiding Overlays.....	380
Deleting Overlays.....	380
Stacking Maps	381
 Chapter 18 Grid Operations	383
Introduction to Grid Operations.....	383
Spline Smoothing.....	383
Expanding a Grid	385
Recalculating a Grid	386

Table of Contents

Grid Filtering	388
The Neighborhood	388
The Digital Filtering Dialog	389
Linear Convolution Filters	390
Nonlinear Filters	393
Edge of Grid	396
Blanked Nodes	398
Grid Filter References	398
Grid Mosaic	399
Grid Mosaic Dialog	399
Resample Methods	401
Joining Grids	402
Mosaic Tips	402
Blanking a Grid File	403
Blanking Inside or Outside a Polygon	404
Blanking Outside Multiple Polygons	405
Blanking Individual Grid Nodes	405
Grid Node Editor	406
The Options Menu	406
Using the Grid Node Editor	408
Grid Convert	409
Converting Grid File Formats	410
Grid Extract	411
The Extract Grid Dialog	412
Extracting Grids	412
Grid Transform	412
The Transform Dialog	412
Transforming a Grid File	415
Grid Math	416
Using the Grid Math Command	416
Using the Grid Math Command to Generate an Isopach Map	417
Using Grid Math to Eliminate Negative Values from a Grid File	418

Table of Contents

Grid Calculus	418
Using Grid Calculus	421
Directional Derivatives	421
Terrain Modeling	427
Differential and Integral Operators	433
Fourier and Spectral Analysis	437
Grid Calculus References	442
Chapter 19 Volumes, Areas, Cross Sections, and Residuals	443
Introduction to Volumes, Areas, Cross Sections, and Residuals	443
Volumes and Areas	443
Volume Accuracy	444
The Grid Volume Report	445
Net Volume	446
Planar Area Calculations	448
Surface Area Calculations	449
Volume Reference	449
Calculating Volumes and Areas	449
Calculating the Volume between Two Surfaces	451
Calculating the Volume of a Lake	452
Volume of a Blanked Grid	453
Cross Sections	454
Cross Section Data File	454
Cross Section Blanking File	455
Blanking Files	455
Creating Cross Section Data	455
Plotting Cross Sections	456
Residuals	457
Calculating Residuals	458
Calculating the Value of the Surface at a Point	459

Chapter 20 Importing and Exporting Maps and Graphics	461
Introduction to Importing and Exporting	461
Importing Objects into Surfer	461
Import vs. Base Map	461
Import	461
Import Options	462
Paste Special	462
Exporting Objects	463
Export Options	463
Specifying File Type During Export	463
Bitmap Export	463
Coordinates and Scaling During Export	465
Spatial Formats	466
Suggestions on Selecting Export Formats	466
 Chapter 21 Creating, Selecting, and Editing Objects.....	467
Introduction to Creating, Selecting, and Editing Objects	467
Creating Objects	467
Text	468
Polygon	469
PolyLine	470
Symbol	471
Rectangle	472
Rounded Rectangle	473
Ellipse	474
Selecting Objects	475
How to Select Objects	475
Bounding Boxes	477
Selection Handles	477
Deselecting Objects	478
Object Properties	478
Modeless Property Dialog	478

Table of Contents

Setting the Properties for a Single Object.....	478
Setting the Properties of Multiple Selections.....	479
Setting Fill, Line, Symbol, and Text Properties.....	479
Label Properties.....	480
Custom Line, Fill, and Color.....	481
Setting Default Properties.....	481
Reshape.....	481
Arranging, Combining, Rotating, and Aligning Objects.....	482
Changing the Order in which Objects are Drawn.....	482
Changing the Order in which Maps are Drawn.....	482
Combining Objects.....	483
Rotating Objects.....	483
Aligning Objects.....	484
Object IDs.....	485
Assigning an ID to a Map.....	485
Deleting Objects.....	485
Moving and Resizing Objects with the Mouse.....	486
Using the Keyboard to Simulate Mouse Operations.....	487
Advanced Transformations.....	488
Transform Dialog.....	488
Transform References.....	489
Using the Object Manager.....	490
Opening and Closing the Object Manager.....	490
Changing the Object Manager Location.....	490
Object Visibility.....	490
Object Manager Tree.....	491
Selecting Objects.....	491
Opening Object Properties.....	491
Editing Object IDs.....	491
Arranging Objects.....	492
Deleting Objects.....	492

Chapter 22 Automating Surfer	493
Introduction to Automation	493
Surfer Automation Objects	494
The Surfer Object Model	494
Overview of Surfer Objects	499
A Brief Introduction to the Major Surfer Objects	502
Surfer Version 6 Object Model	508
Introducing Scripter	508
Scripter Windows	509
Working with Scripts	511
Using Scripter Help	511
Writing Scripts	512
The Scripter BASIC Language	513
Code, Class, and Object Modules	524
Creating Dialog Boxes	528
Type Library References	532
The Object Browser	533
Compatibility with Visual BASIC	534
Suggested Reading	535
Running Scripts	535
Running Scripts from Scripter	535
Running Scripts from the Command Line	536
Debugging Scripts	536
Viewing Errors	536
Run-Time Errors	537
Script Runs Incorrectly	537
Debug.Print	537
Stop or Pause	538
Viewing Variable Values	538
Changing Variable Values	538
Step	538
Breakpoint	539

Table of Contents

Trace	540
Stack	540
Modules	540
Examples	540
Creating and Printing a Contour Map	540
Opening, Saving, and Closing Documents	542
Creating a Variogram	543
Overlaying Maps	545
Modifying Axes	546
Chapter 23 Preferences.....	547
Introduction to Preferences	547
General	547
Drawing	549
Selection Group	549
Default Attributes Group	550
Miscellaneous Options.....	550
Rulers and Grid.....	551
Dialog Defaults	552
Formats for Attribute Values	554
Colors.....	554
Line Properties.....	555
Fill Properties	555
Font Properties.....	556
Symbol Properties.....	557
Numeric Label Properties	558
Using Custom Setting Files	559
Appendix A Mathematical Functions.....	561
Introduction to Mathematical Functions	561
Formula Calculation	561
Mathematical Functions.....	562

Table of Contents

Trigonometric Functions.....	562
Bessel Functions	562
Exponential Functions	563
Miscellaneous Functions.....	563
Statistical Functions of an Interval of Columns.....	564
Boolean Expressions.....	564
Transforming Data in the Worksheet.....	564
Errors	566
Examples	567
 Appendix B Math Text Instructions.....	569
Introduction to Math Text.....	569
Math Text Instruction Syntax	569
Math Text Instructions.....	570
Instructions that Change Typefaces, Sizes, and Styles	570
Instructions that Change Text Color	571
Instructions that Change Text Position	571
Instructions Used to Insert Special Characters or Date and Time.....	572
Examples of Math Text Instructions.....	573
 Appendix C File Formats	575
Introduction to File Formats	575
Grid File Formats.....	575
GS ASCII Grid File Format.....	575
GS Binary Grid File Format	578
Surfer Grid and Surfer 7 Grid File Format	579
Level File Format	582
Color Spectrum File Format	583
Import and Export File Formats.....	585
Golden Software Boundary Files.....	586
Golden Software Blanking Files.....	586
Golden Software PlotCall Files	587

Table of Contents

Golden Software Interchange Files.....	588
USGS SDTS Topological Vector Profile Files.....	588
USGS Digital Line Graph Files.....	588
Bitmaps.....	588
Metafiles.....	589
Atlas Boundary Files.....	590
ESRI ArcInfo Export Format Files.....	592
ESRI Shapefiles.....	592
MapInfo Interchange Format Files.....	592
AutoCAD DXF Files.....	592
Data File Formats.....	593
ASCII Data File Formats.....	593
Excel Data Files.....	594
SYLK Files.....	594
Lotus Data Files.....	594
Appendix D Line Styles, Fill Patterns, and Colors	595
Introduction to Line Styles, Fill Patterns, and Colors.....	595
Line Styles.....	595
Fill Patterns.....	596
Colors.....	596
Index	599

Chapter 1

Introducing Surfer

Introduction to Surfer

Surfer® is a grid-based graphics program. **Surfer** interpolates irregularly spaced XYZ data into a regularly spaced grid. Grids may also be imported from other sources, such as the United States Geological Survey (USGS). The grid is then used to produce different types of maps including contour, vector, wireframe, image, shaded relief, and surface maps. Many gridding and mapping options are available allowing you to produce the map that best represents your data.

Maps can be displayed and enhanced in **Surfer** by the addition of boundary information, posting data points, combining several maps, adding drawings, and annotating with text.

An extensive suite of gridding methods is available in **Surfer**. The variety of available methods provides different interpretations of your data, and allows you to choose the most appropriate method for your needs. In addition, data metrics allow you to gather information about your gridded data.

The grid files themselves can be edited, combined, filtered, sliced, queried, and mathematically transformed. For example, a color-filled contour map can be drawn from a grid of groundwater surface elevations. This grid can then be numerically differentiated, and a gradient-based vector map subsequently generated. The vector map can then be drawn as an overlay on the contour map. The resulting graphic shows not only the contours, but also the flow paths.

Surface area, projected planar area, and volumetric calculations can be performed quickly in **Surfer**. Cross-sectional profiles can also be computed and exported.

The **Scripter**™ Program, included with **Surfer**, is useful in creating, editing, and running script files that automate **Surfer** procedures. By writing and running script files, simple mundane tasks or complex system integration tasks can be performed precisely and repetitively without direct interaction. **Surfer** also supports ActiveX Automation using any compatible client, such as Visual BASIC. These two automation capabilities allow **Surfer** to be used as a data visualization and a map generation post-processor for any scientific modeling system.

New Features

General

- **Surfer** is designed for Windows 98, Me, 2000, and XP.
- A **Help | Check for Update** command is available to easily update **Surfer**.
- A new HTML-based help system allows advanced searching options.
- We have added several new scripting examples.

User Interface Features

- Bitmaps, image maps, and shaded relief maps can be rotated and transformed.
- Property dialogs are modeless. This means you can open a property dialog and continue working with **Surfer**. The property dialog updates when the selection changes.
- You can rotate, tilt, and change the field of view with the **Map | Trackball** command.
- **Edit | Paste** and **Edit | Paste Special** automatically pastes the object in the center of the page.
- The **View | Pan Realtime** allows you to easily move the contents of the plot window.
- The **Zoom Realtime** tool allows you to zoom in and out by dragging the mouse.
- Map scale, view, limits, and background have been moved to the map property dialog for easier access.
- Bitmaps are now compressed in [.SRF] files, resulting in smaller files and faster saving and loading times.
- Create maps quickly by selecting the map tool. The default map properties are used and these properties can be edited by opening the new property dialog.
- Use the **Help | Feedback** command to send a problem report, suggestion, or information request by e-mail.
- Multithreading allows for a smoother redraw and more responsive user interface.
- The arrow keys can now be used to "nudge" the selected objects one pixel at a time for precise positioning.

Surfaces

- 3D rendered surfaces have been added to **Surfer**.
- All planar map types and other surfaces can be overlaid on surfaces.
- Bitmaps can be overlaid on surfaces.
- Control the color, lighting, mesh, base, and overlay color blending of a surface.
- Added many new color spectrum files including, ChromaDepth.clr and Terrain.clr.

Gridding

- Filter grids using over 60 pre-defined filters or define your own filter.
- Load large data files faster when gridding (there is no pre-sorting or pre-duplicate checking).
- Use *Data Metrics* to gather information about the data and create a grid of this information.
- The **Grid | Mosaic** command combines multiple adjacent or overlapping grids.
- Use the new *Cross Validate* option to assess the quality of the selected gridding method.
- A map of the Delaunay triangulation can be created when gridding with the Triangulation with Linear Interpolation or Natural Neighbor methods.
- Added Moving Average and Local Polynomial gridding methods.
- Faults are retained in grids after blanking.
- Added Cubic and Pentaspherical variogram models.


Volume

Added a Z scale factor to volume calculations. This allows you to set a scaling factor when your XY units are not the same as the Z units.

Digital Elevation Models

- **Surfer** provides native support for the newest SDTS format.
- Digital Terrain Elevation Data (DTED) files can be used where grids are needed in the program.

Reports

- New reporting functionality has been added to grid info , **Grid | Data**, **Grid | Volume**, and **Grid | Variogram**.
- Reports now include numerous additional statistics on the data.

Import / Export

- A new set of import filters include Enhanced Metafile, Golden Software Interchange, and ESRI ArcInfo Export Format.
- A new set of export filters include Enhanced Metafile, MapInfo Interchange Format, Golden Software Interchange, and Golden Software Boundary.

Chapter 1 - Introducing Surfer

Page Setup

- The 32-inch page limit has been eliminated under all operating systems.
- The resolution of graphics operations has been increased by 1000 times, resulting in more accurate output when zoomed in, and when the graphics are exported.




Worksheet

Up to one billion rows and columns are available in the worksheet.

Screen Layout and Window Types

Surfer contains three document window types, the plot window, worksheet window, and grid node editor window. Maps are displayed and created in the plot window. The worksheet window is used to display, edit, transform, and save data in a tabular format. The grid node editor window is used to display and edit Z values for the selected grid.

Opening Windows

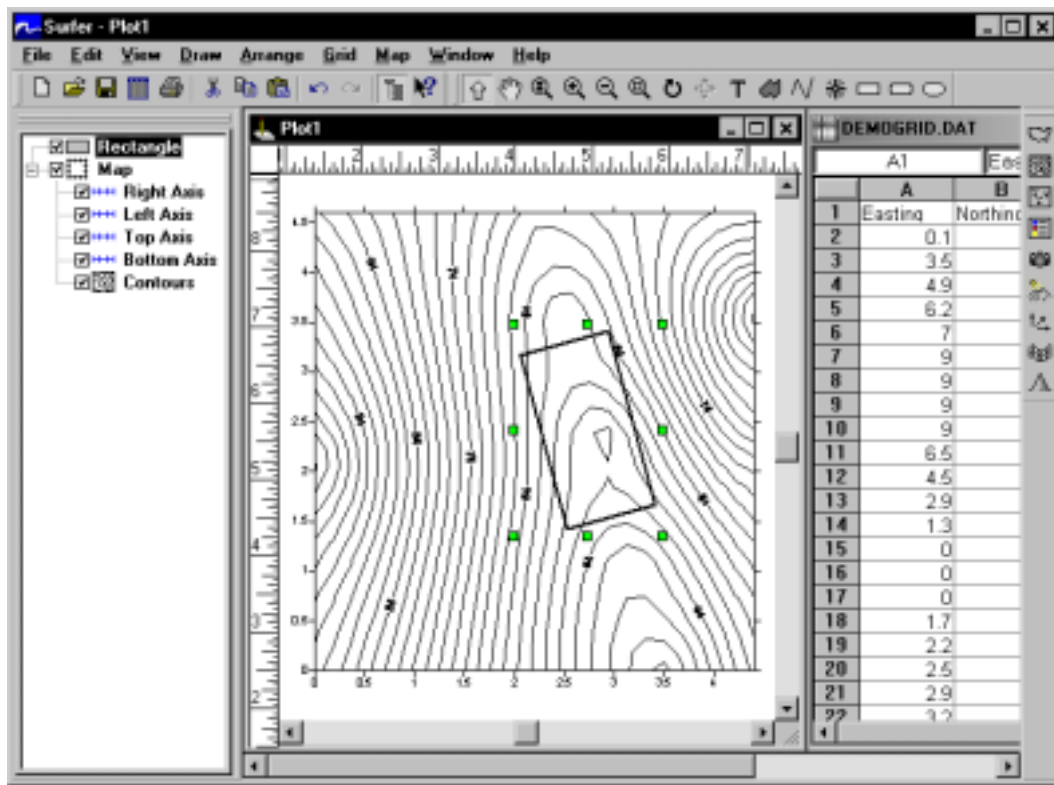
Selecting the **File | Open** command, or clicking the  button opens any of the three window types, depending on the type of file selected. The **File | New** command, or clicking the  button creates a new plot window or worksheet window. To open a new worksheet window directly, you can click the  button.

Object Manager

When **Surfer** starts, the **Object Manager** is visible in the plot window by default. It contains a hierarchical list of the objects in the **Surfer** plot window. The **Object Manager** is initially docked at the side of the window, giving the window a split appearance; however, it can be dragged and placed anywhere on the screen.

Toolbars

All window types in **Surfer** include toolbars that contain buttons for many common commands. The toolbars are initially docked, but they too can be dragged and placed anywhere on the screen.



*This is the **Surfer** window with the **Object Manager** is on the left, the plot window in the center, and the worksheet window is on the right.*

Scripter

Tasks can be automated in **Surfer** using Golden Software's **Scripter** program or any ActiveX Automation-compatible client, such as Visual BASIC. A script is a text file containing a series of instructions for execution when the script is run. **Scripter** can be used to perform almost any task in **Surfer**. For more information, see *Chapter 22, Automating Surfer* in the User's Guide or use the **Help | Automation Help** command.

Using Surfer

The general steps to progress from a XYZ data set to a finished, grid-based map are as follows:

1. Create a XYZ data file. This file can be created in a **Surfer** worksheet window or outside of **Surfer** (using an ASCII text editor, for example).
2. Create a grid file [.GRD] from the XYZ data file using the **Grid | Data** command.
3. To create a map, select the map type from the **Map** menu and use the grid file from step two. Grid-based maps include contour, image, shaded relief, vector, wireframe, and surface maps.
4. Use **File | Save** to save the map as a **Surfer** file [.SRF] that contains all of the information needed to recreate the map, including the data file.

File Types

Surfer uses four basic file types: data, grid, boundary, and **Surfer** files [.SRF].

Data Files

Data files contain the input data provided by the user, and are used to produce grid files, plot data points on a map, or generate a residuals log. These files are generally referred to as "XYZ data files" or "data files" throughout the documentation. Data can be read from various file types, and must contain numeric XY location coordinates as well as numeric Z values. The Z values contain the variable to be modeled, such as elevation, concentration, rainfall, or similar types of values.

Grid Files

Grid files are used to produce several different types of grid-based maps, to perform calculations such as volume, residuals, and grid math, and to carry out blanking, smoothing, and slice operations. Grid files contain a regularly spaced rectangular array of Z data organized in columns and rows. **Surfer** can also use USGS digital elevation models and DTED files to perform most of the operations that can be performed with grid files.

Boundary Files

Boundary files contain XY areal data such as state boundaries, rivers, or point locations. Boundary files are used to overlay a base map on another map, or to specify the boundary limits for blanking, faults, breaklines, and slice calculations. Boundary files can be vector files, metafiles, or bitmap files.

Surfer Files

Surfer files [.SRF] preserve all the objects and object settings contained in a plot window. These files are called **Surfer** files [.SRF] throughout the documentation.

Gridding

Gridding is the process of taking irregularly spaced XYZ data and producing a grid file that contains a *regularly* spaced array of Z data. **Surfer** has several different gridding methods. These gridding methods define the way in which the XYZ data are interpolated when producing a grid file. A mathematical formula can also be used to create a grid file. For more information, see *Chapter 4, Creating Grid Files*.

Grid Utilities

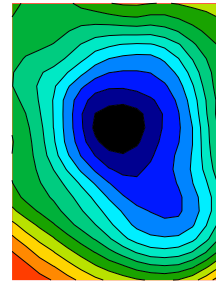
There are many ways to manipulate grid files in **Surfer**. The **Grid** menu contains several utilities used to convert, combine, mirror, limit, slice, rotate, and extract grids. In addition, volume calculations, smoothing, blanking, cross section creation, and residual calculations can all be performed in **Surfer** using these utilities. For more information, see *Chapter 18, Grid Operations*.

Map Types

Several different map types can be created, modified, and displayed with **Surfer**. These map types include contour, base, post, image, shaded relief, vector, wireframe, and surface maps.

Contour Map

A contour map is a two-dimensional representation of three-dimensional data. Contours define lines of equal Z values across the map extents. The shape of the surface is shown by the contour lines. Contour maps can display the contour lines; they can also display colors and patterns between the contour lines.

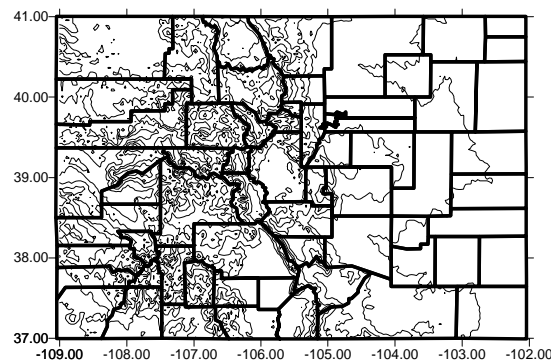


This is a filled contour map showing different colors for various Z levels.

Chapter 1 - Introducing Surfer

Base Map

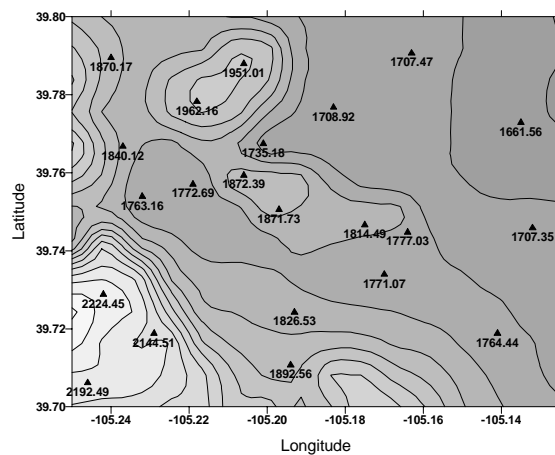
Base maps display boundaries on a map. Boundaries can include roads, buildings, streams, lakes, and so on. Base maps can be produced from several file formats. Refer to *Appendix C, File Formats* for more information.



A base map, consisting of Colorado county boundary lines, is displayed on top of a contour map of Colorado.

Post Maps

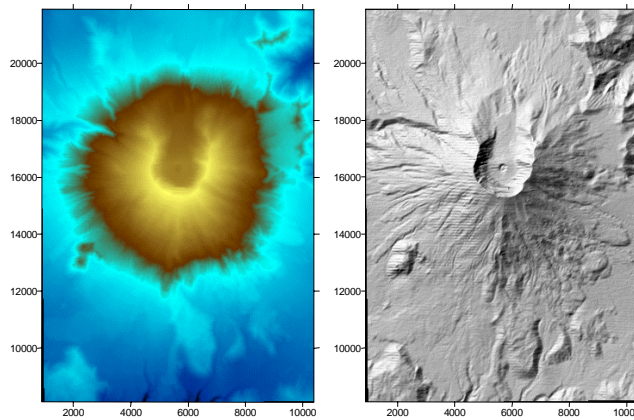
Post maps and classed post maps show data locations on a map. You can customize the symbols and text associated with each data location on the map.



This is a post map combined with a filled contour map.

Image Maps and Shaded Relief Maps

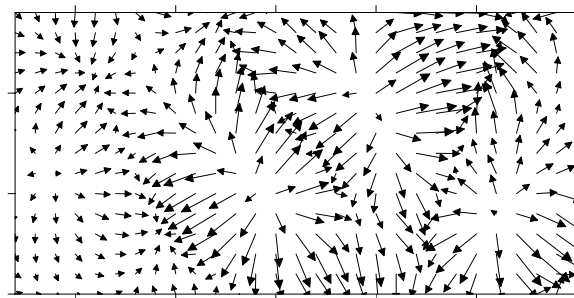
Image maps and shaded relief maps are raster images based on grid files. Image maps assign colors based on Z values from a grid file. Shaded relief maps assign colors based on slope orientation relative to a light source.



The same DEM file was used to create the image map on the left and the shaded relief map on the right.

Vector Maps

Vector maps display direction and magnitude data using individually oriented arrows. For example, at any grid node on the map, the arrow points in the direction of steepest descent ("downhill") and the arrow length is proportional to the slope magnitude. In **Surfer**, vector maps can be created using the information in one grid file (i.e. a numerically computed gradient) or two different grid files (i.e. each grid giving a component of the vectors).

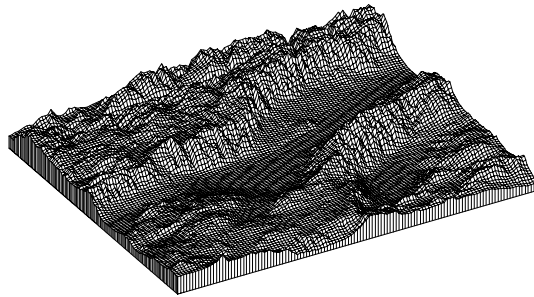


This is an example of a vector map.

Chapter 1 - Introducing Surfer

Wireframes

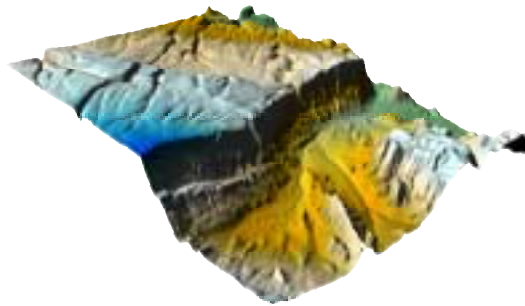
Wireframes are three-dimensional representations of a grid file. Lines of equal Z values can be displayed on a wireframe, creating a stacked contour map.



A wireframe displays a grid's rows and columns.

Surfaces


Surfaces are color three-dimensional representations of a grid file. The colors, lighting, overlays, and mesh can be altered on a surface.



Surfaces show grids in 3D color.

Reports

You can gather information about your data by:

- clicking the *Statistics* button in the **Grid Data** dialog to obtain a Data Statistics Report,
- checking the *Grid Report* option in the **Grid Data** dialog to create a Gridding Report,
- clicking the *Report* button in the **Variogram Properties** dialog to create a Variogram Grid Report,
- checking the *Generate Report* option in the **New Variogram** dialog to create a Variogram Grid Report,
- clicking the  button in a map properties dialog to generate Grid Information,
- or checking the *Report* option in the **Cross Validation** dialog to create a Cross Validation Report.

If you make changes to the data selection (e.g. changing a data column or changing the data filtering method), you can generate a new report by repeating one of the processes listed above. Review the *Reports* topic in online help for details on the report contents.

System Requirements

- Operating System: Windows 98, Me, XP, 2000, or higher
- Monitor Resolution: 800 X 600 X 256 colors minimum
- Hard Disk: At least 25 MB of free hard disk space
- RAM: 32 MB free minimum, 64 MB free recommended
- CPU: 100 MHz Pentium processor minimum

A Note about the User's Guide and Online Help

Various font styles are used throughout the **Surfer** User's Guide and online help. **Bold** text indicates menu commands, dialog names, and page names. *Italic* text indicates items within a dialog such as group box names, options, and field names. For example, the **Import File** dialog contains a *Look in* drop-down list. Bold and italic text may occasionally be used for emphasis.

Also, menu commands appear as **Map | Base Map**. This means, "click on the **Map** menu at the top of the plot window, then click on **Base Map** within the **Map** menu list." The first word is always the menu name, followed by the commands within the menu list.

Installation Directions

Golden Software does not recommend installing **Surfer 8** over any previous version of **Surfer**. **Surfer** version 4 (DOS), 5, 6, 7, and 8 can coexist on the same computer as long as they are installed in separate directories.

To install **Surfer 8** on Windows XP or 2000, you need to have administrator rights for that computer.

To install **Surfer**:

1. Insert the **Surfer** CD in the CD-ROM drive. The install program automatically begins on most computers.
2. Choose **Install Surfer** from the **Setup Options** window to begin the installation.

If the installation does not begin automatically, double-click on setup.exe in Windows Explorer.

Network Installation

An administrative install may be performed in order to install files to a network server. Once installed on the network server, individual workstation installations can be performed. The server software must support long file names.

ATTENTION: You may use **Surfer** on a networked system if the number of **Surfer** users on the network at one time does not exceed the number of licensed copies of **Surfer**.

To install **Surfer** on the server:

1. Log on to the file server with administrator rights.
2. Click **Start | Run**.
3. Enter the path to SETUP.EXE followed by /a. For example, r:\Setup.exe /a
4. When setup asks for a destination folder, choose one on the file server (e.g. C:\SurferServer). This should be a new, empty directory. Setup copies all the **Surfer** files plus the setup program and its associated files to the server drive.
5. We recommend flagging the server folder contents as read-only.

NOTE: If you wish to run **Surfer** on a Windows server itself, you will need to run setup again without the /a switch and install **Surfer** normally to a new directory.

Next, install **Surfer** at each workstation:

1. Log on to the file server from each workstation that will run **Surfer**.
2. Start Windows and run the copy of SETUP.EXE located in the server folder. **DO NOT RUN SETUP FROM THE CD-ROM!**
3. When the setup asks for a destination folder, choose one on the local hard drive.

You will need to enter a **Surfer** serial number the first time **Surfer** is run on each workstation.

Troubleshooting Installations

If you are experiencing errors with the installation, please install **Surfer** with error logging.

To create an installation error log:

1. In Windows, go to **Start | Run**.
2. Type [drive letter]:\[path]\setup.exe" /V"/L* [drive letter]:\[path]\S8.log
Replace the [drive letter] and [path] with a drive letter and path on your computer. The path to the log file cannot contain spaces.
3. Click OK. After the installation process terminates, the log file is created in the path you specify. Please send the log file to technical support along with the installation message text.

Updating Surfer

To update your version of **Surfer**, open the program and select **Help | Check for Update**.

Do not install a newer FULL version of **Surfer 8** over a previous version of **Surfer 8**. This does not update the software. (i.e. Do not install 8.02 over 8.00.) If for some reason you need to do a full installation of **Surfer 8**, uninstall the previous version before installing the updated version.

Uninstalling Surfer


To uninstall **Surfer**, use Add/Remove Programs in the Control Panel. You can access the Control Panel by clicking the Windows Start button, clicking on Settings, and then clicking on Control Panel.


Getting Help

Online Help

The User's Guide is just one part of the documentation for **Surfer**. Extensive additional information about **Surfer** is located in the online help. To access online help, choose **Help | Contents** for a hierarchical arrangement of help topics. By clicking on the topic of interest, information regarding the topic appears.

Context-Sensitive Help

Surfer also contains context-sensitive help. Highlight a menu command, window region, or dialog box, press the F1 key, and help is displayed for the highlighted item. Another way to access context-sensitive help is by clicking on the context-sensitive help button. After clicking the  button, the cursor appears with a ? next to it. Simply select the item for which help is desired with the modified pointer and a help window appears.

In addition, the dialog boxes contain a help button. When you have an open dialog, click the  button in the dialog title bar to obtain help for that dialog. Many dialog details are contained in online help.

Internet

There are several Web resources available for help.

- Click the *Forum* button in online help to research a **Surfer** question or to post a question.
- You can use the **Help | Feedback** command to send a problem report, suggestion, or information request by e-mail.

Technical Support

Golden Software's technical support is free to registered users of Golden Software products. Our technical support staff is trained to help you find answers to your questions quickly and accurately. We are happy to answer any of your questions about any of our products, both before and after your purchase. We also welcome suggestions for improvements to our software and encourage you to contact us with any ideas you may have for adding new features and capabilities to our programs.

Chapter 1 - Introducing Surfer

Technical support is available Monday through Friday 8:00 AM to 5:00 PM Mountain Time, excluding major United States holidays. We respond to e-mail and fax technical questions within one business day. When contacting us with your question, have the following information available:

- Your **Surfer** serial number (located in the front cover of this User's Guide or in **Help | About Surfer**)
- Your **Surfer** version number, found in **Help | About Surfer**
- The operating system you are using (Windows 98, Me, XP, or 2000)
- The exact wording of the first error message that appears (if any)

Contact Information

Telephone: 303-279-1021

Fax: 303-279-0909

E-mail: surfersupport@goldensoftware.com

Web: www.goldensoftware.com (includes FAQs and support forum)

Mail: Golden Software, Inc., 809 14th Street, Golden, Colorado, 80401-1866, USA

Chapter 2

Tutorial

Tutorial Introduction

This tutorial is designed to introduce you to some of **Surfer's** basic features. After you have completed the tutorial, you should be able to begin creating your own grids and maps.

Tutorial Lessons

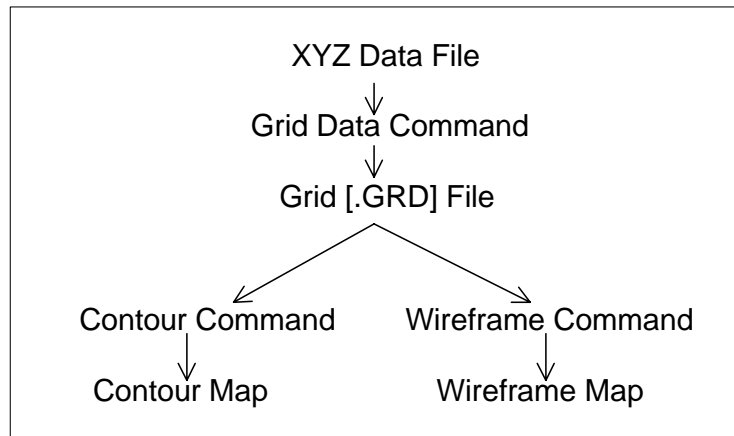
The following is an overview of lessons included in the tutorial.

- *Lesson 1 - Creating an XYZ Data File* shows you how to import a data file and how to create a new data file.
- *Lesson 2 - Creating a Grid File* shows you how to create a grid file, the basis for most map types in **Surfer**.
- *Lesson 3 - Creating a Contour Map* shows you how to create a contour map and change the contour map properties.
- *Lesson 4 - Creating a Wireframe* shows you how to create a wireframe and set the wireframe properties.
- *Lesson 5 - Posting Data Points and Working with Overlays* shows you how to create a post map and overlay it with a contour map so that both maps share the same axes.
- *Lesson 6 - Introducing Surfaces* shows you some of the features associated with the new surface maps.

The lessons should be completed in order, however, they do not need to be completed in one session.

Understanding How to Use Surfer

The most common application of **Surfer** is to create a grid-based map from an XYZ data file. The **Grid | Data** command uses an XYZ data file to produce a grid file. The grid file is then used by most of the **Map** menu commands to produce maps. Post maps and base maps do not use grid files.



The flow chart to the right illustrates the relationship between XYZ data files, grid files, contour maps, and wireframes.

This flow chart illustrates the relationship between XYZ data files, grid files, contour maps, and wireframes.

Starting Surfer

To begin a **Surfer** session:

1. Click on the Windows **Start** button.
2. Navigate to **Programs | Golden Software Surfer 8** and click **Surfer 8**.
3. **Surfer** starts with a new empty plot window. This is the work area where you can produce grid files, maps, and modify grids. If this is the first time that you have opened **Surfer**, you will be prompted for your serial number. Your serial number is located on the inside front cover of this User's Guide.

Lesson 1 - Creating an XYZ Data File

An XYZ data file is a file containing at least three columns of data values. The first two columns are the X and Y coordinates for the data points. The third column is the Z value assigned to the XY point. Although it is not required, entering the X coordinate in Column A, the Y coordinate in Column B, and the Z value in Column C is a good idea. **Surfer** looks for these coordinates in these columns by default.


	A	B	C
1	X Data	Y Data	Z Data
2	0.1	0	90
3	9	3	48
4	1.3	7	52
5	4.7	1	66
6	1.7	5.6	75
7	6	1	50
8	2.5	3.6	60

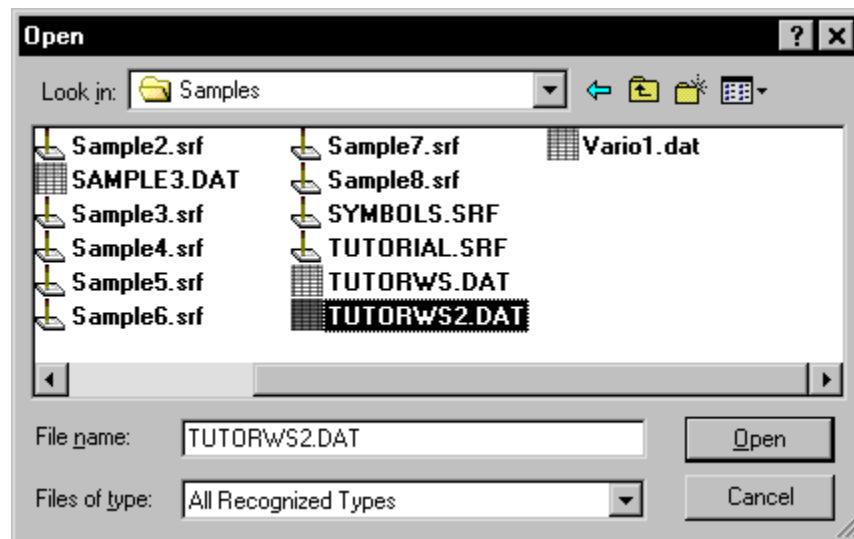
This is a simple XYZ data file. Notice that the X, Y, and Z data are placed in Columns A, B, and C, respectively.

See *Chapter 3, Data Files and the Worksheet* for a complete description of XYZ data files, worksheet windows, and how to manipulate data in the worksheet.

Opening an Existing Data File

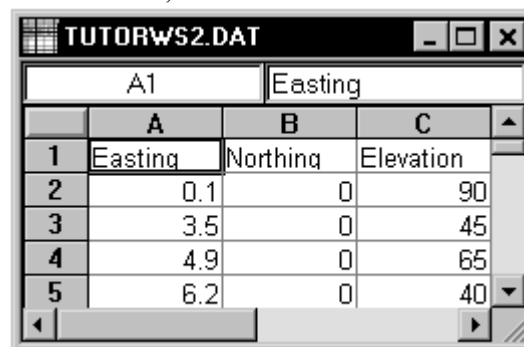
To look at an example of an XYZ data file, open TUTORWS2.DAT into a worksheet window:

1. Choose the **File | Open** command, or click the  button to select the XYZ data file to display in the worksheet window.



*In the **Open** dialog, select the sample file TUTORWS2.DAT included with Surfer.*

2. Double-click on the SAMPLES folder. In the list of files, click TUTORWS2.DAT and then click the *Open* button to display the file in the worksheet window.
3. Notice that the X coordinate (Easting) is in Column A, the Y coordinate (Northing) is in Column B, and the Z value (Elevation) is in Column C. Although it is not required, the header text (the text in row 1) is helpful in identifying the type of data in the column, and this information is used in dialog boxes when selecting worksheet columns.




TUTORWS2.DAT			
A1		Easting	
	A	B	C
1	Easting	Northing	Elevation
2	0.1	0	90
3	3.5	0	45
4	4.9	0	65
5	6.2	0	40

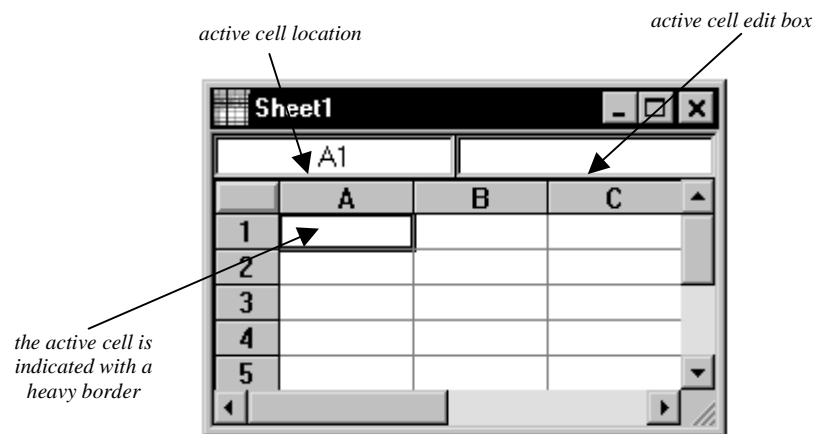
When a data file is displayed, the name of the file is shown in the title bar of the worksheet window. In this file, row 1 contains descriptive information about each column of data.

Creating a New Data File

The **Surfer** worksheet can also be used to create a data file.

To open a blank worksheet window and begin entering data:

1. Choose the **File | New** command or click the  button.
2. Click the *Worksheet* option in the **New** dialog and then click the OK button to display a new empty worksheet window.




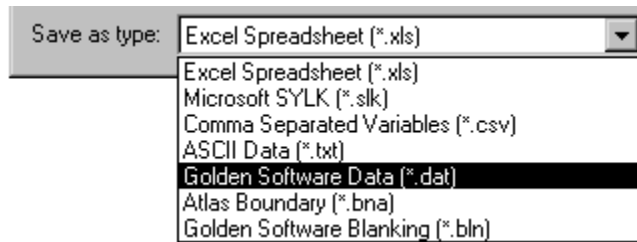
Data are entered into the active cell of the worksheet.

3. The active cell is selected by clicking on the cell or by using the arrow keys to move between cells. The active cell is indicated by a heavy border and the contents of the active cell are displayed in the active cell box.
4. When a cell is active, enter a value or text, and the information is displayed in both the active cell and the active cell box.
5. The BACKSPACE and DELETE keys can be used to edit data as you type.
6. Press the ENTER key. The data are entered into the cell and the active cell moves down one row.

Saving the Data File

When you have completed entering all of the data:

1. Choose the **File | Save** command, or click the  button. The **Save As** dialog is displayed if you have not previously saved the data file.
2. In the *Save as type* list, choose the *Golden Software Data (*.dat)* option.
3. Type the name of the file into the *File name* box.
4. Click *Save* and a **GSI Data Export Options** dialog opens.
5. Accept the defaults in the **GSI Data Export Options** dialog by clicking the OK button.



Click the down arrow to select Golden Software Data (.dat) for the file type.*

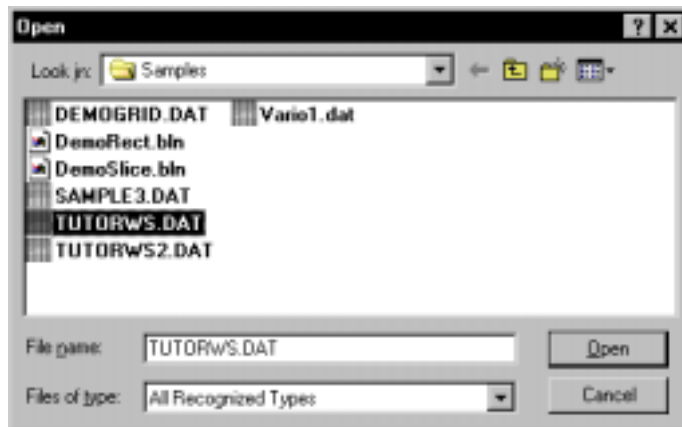
The file is saved in the Golden Software Data [DAT] format with the file name you specified. The name of the data file appears at the top of the worksheet window.

Lesson 2 - Creating a Grid File

Grid files are required to produce a grid-based map. Grid-based maps include contour maps, image maps, shaded relief maps, 1-grid vector maps, 2-grid vector maps, wireframes, and surfaces. Grid files are created using the **Grid | Data** command. The **Data** command requires data in three columns, one column containing X data, one column containing Y data, and one column containing Z data. We have included a sample XYZ data file (TUTORWS.DAT) with **Surfer** for you to see how to produce a grid file. After completing the tutorial, if you need to produce an XYZ data file of your data for your work, see *Creating a New Data File* on page 21.

To produce a grid file from TUTORWS.DAT:

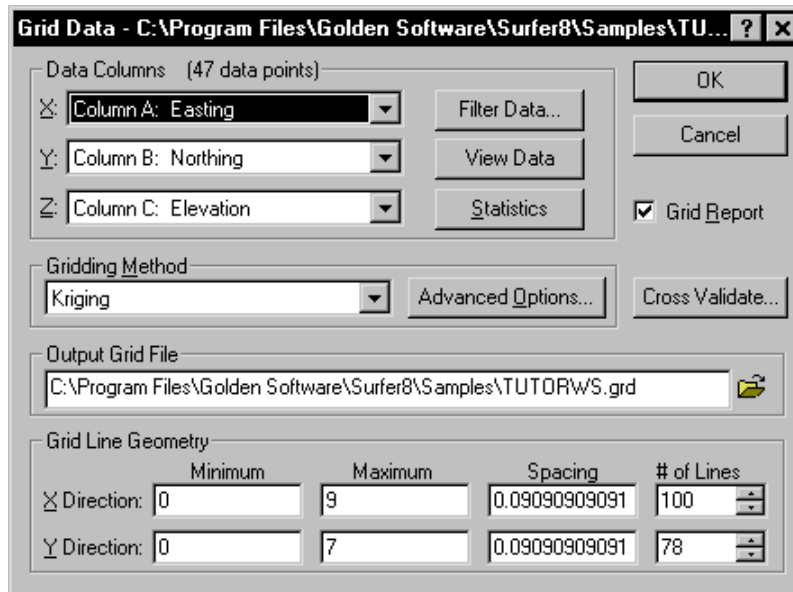
1. If you have the worksheet window open, click on the **Window** menu and choose **Plot1**.
Alternatively, you can create a new plot window with **File | New** (select *Plot Document* and then click OK).
2. Choose the **Grid | Data** command.
3. In the **Open** dialog, select TUTORWS.DAT (located in **Surfer's** SAMPLES folder). The name appears in the *File name* box below the list of data files.
4. Click **Open** and the **Grid Data** dialog is displayed.
Alternatively, you can double-click the data file name to display the **Grid Data** dialog.
5. The **Grid Data** dialog allows you to control the gridding parameters. Take a moment to look over the various options in the dialog. Do not make changes at this time, as the default parameters create an acceptable grid file.
 - The *Data Columns* group is used to specify the columns containing the X and Y coordinates, and the Z values in the data file.
 - The *Grid Line Geometry* group is used to specify the XY grid limits, grid spacing, and number of grid lines (also referred to as rows and columns) in the grid file.
 - The *Gridding Method* group is used to specify the interpolation method and interpolation options. The *Gridding Method* is *Kriging* by default.
 - The *Output Grid File* group is used to specify the path and file name for the grid file.



Select TUTORWS.DAT from **Surfer's** SAMPLES folder to create a grid file.

Chapter 2 - Tutorial

- The *Grid Report* option is used to specify whether to create a statistical report for the data.



*The **Grid Data** dialog allows you to specify the gridding parameters to use when creating the grid file.*

6. Click OK. In the status bar at the bottom of the window, a display indicates the progress of the gridding procedure. By accepting the defaults, the grid file uses the same path and file name as the data file, but the grid file has a [.GRD] extension.




The status bar indicates the progress of the gridding procedure.

7. By default, a message appears after gridding the data. Click OK in the grid file has been created message box.
8. If *Grid Report* is checked, a report is displayed. You can minimize or close this report.

Lesson 3 - Creating a Contour Map

The **Map | Contour Map** command creates a contour map based on a grid file.

To create a contour map of the TUTORWS.GRD file created in the previous lesson:

1. Choose the **Map | Contour Map | New Contour Map** command, or click the  button.
2. The **Open Grid** dialog is displayed. The grid file you just created (TUTORWS.GRD) should appear in the *File name* box. If the file does not appear in the *File name* box, select it from the file list.
3. Click the *Open* button to create a contour map.
4. The map is created using the default contour map properties.
5. If you want the contour map to fill the window, choose the **View | Fit to Window** command.

Opening the Map Properties

After creating a map, you can change the map properties. There are several ways to open an object's properties. The most common method is to double-click on the object. Refer to *Object Properties* in *Chapter 21, Creating, Selecting, and Editing Objects* for alternative methods of opening properties.

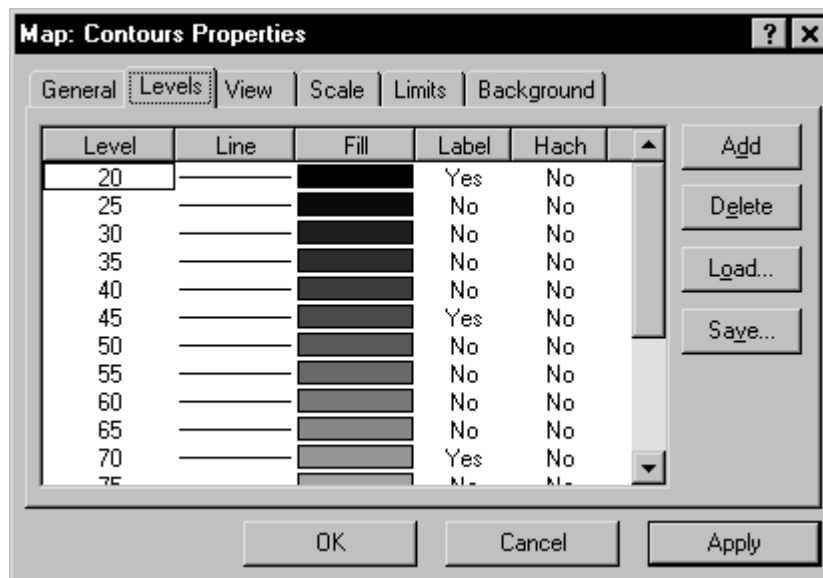
Changing Contour Levels

After you create a contour map, you can easily modify any of the map features. For example, you might want to change the contour levels displayed on the map.

To change the contour levels of the map you just created:

1. Place the pointer inside the limits of the contour map and double-click to open the contour map properties dialog.

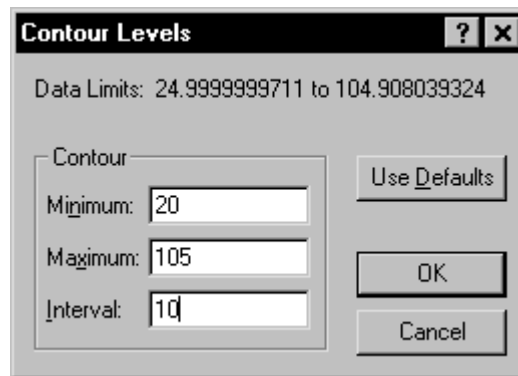
2. In the contour map properties dialog, click the **Levels** page to display the contour levels and contour line properties for the map. In this example, the contour levels begin at $Z = 20$. Click on the scroll bar at the right to scroll to the bottom. You can see that the maximum contour level is $Z = 105$ for this map and that the contour interval is 5.



Click the **Levels** page to display the contour level properties.

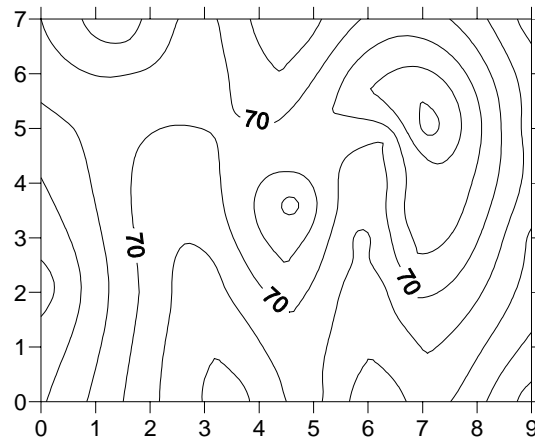
3. To change the contour range and interval, click the *Level* button and the **Contour Levels** dialog is displayed. This shows the *Minimum* and *Maximum* contour level for the map and the contour *Interval*.

4. Double-click in the *Interval* box and type the value 10. Click the OK button and the **Levels** page is updated to reflect the change.



*Open the **Contour Levels** dialog by clicking on the Level button on the **Levels** page.*

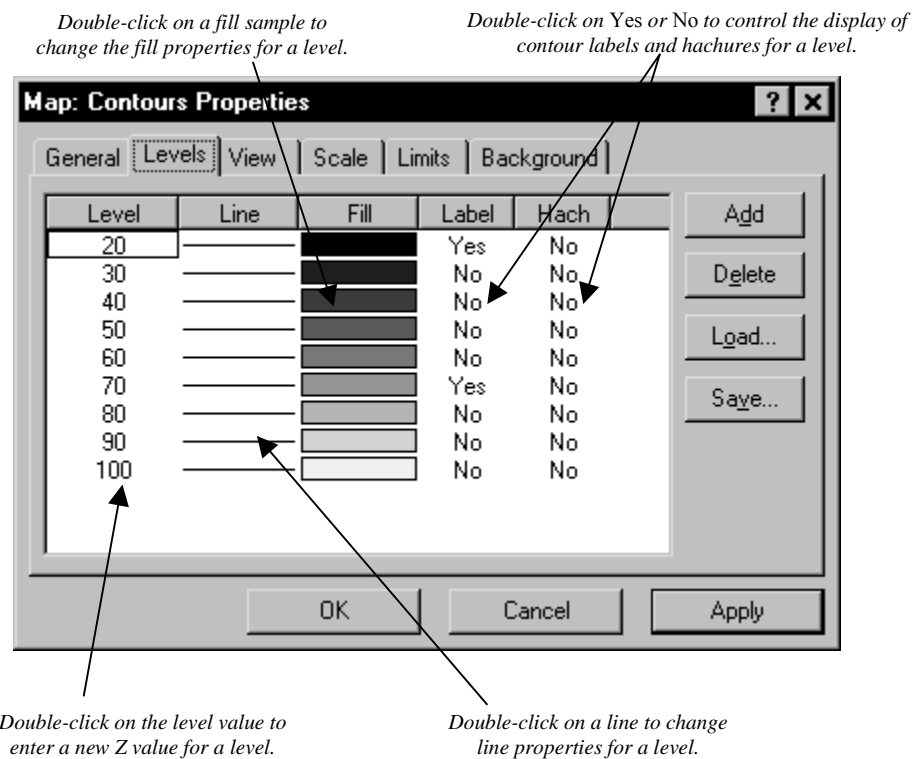
5. Click OK in the contour map properties dialog and the map is redrawn with the new contour levels.



After completing step 5, the contour map is redrawn using 10-foot contour intervals.

Changing Contour Line Properties

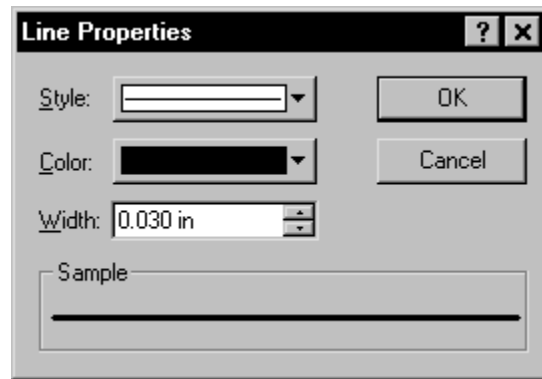
You can double-click any of the elements in the list on the **Levels** page to modify the individual element. For example, you can double-click an individual Z value in the list to change the Z value for that particular contour level. You can also double-click the line style for an individual level to modify the line properties for the selected level. This provides a way to emphasize individual contour levels on the map.



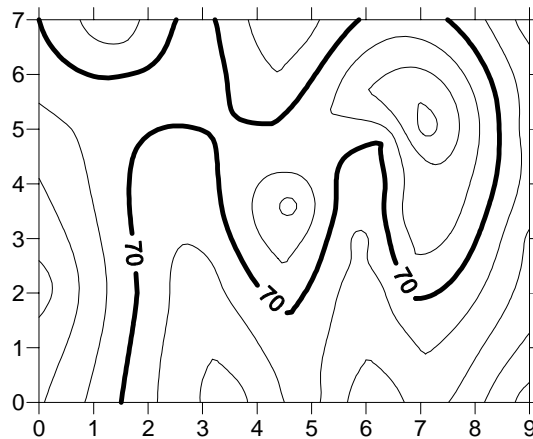
You can double-click on individual elements on the **Levels** page to set specific parameters for the selected level.

To change contour line properties:

1. Double-click the contour map to open the map properties.
2. On the **Levels** page, double-click the line sample for the contour level at $Z = 70$ to open the **Line Properties** dialog.
3. You can select the line color, style, or width for the selected line in the **Line Properties** dialog. In the **Width** box, click the up arrow, and change the width value to 0.030 in. (A width of 0.000 in is equivalent to one pixel width.)
4. Click OK in the **Line Properties** dialog and the **Levels** page is updated to reflect the change.
5. Click OK in the map properties dialog and the map is redrawn. The contour line at $Z = 70$ is drawn with a thicker line.



*You can use the **Line Properties** dialog to set the properties of the selected line.*



The contour line at $Z = 70$ appears bolder than the other contour lines on this map after changing the line properties.

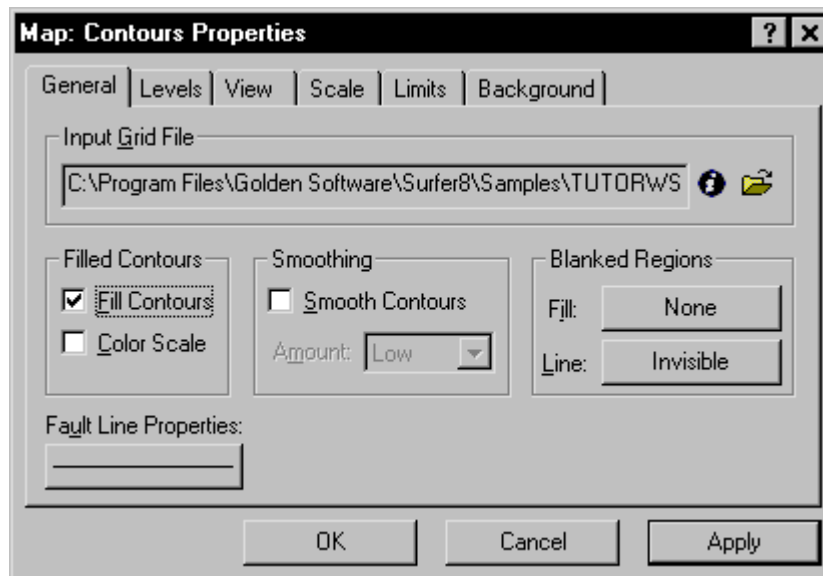
Adding Color Fill between Contour Lines

Color fill can be assigned to individual levels in the same way as line properties. Alternatively, you can assign colors based on a gradational spectrum between two colors.

The **Levels** page in the contour map properties dialog shows a correspondence between a level (under the *Level* button) and a color (under the *Fill* button). The colors are used to fill in the space between the corresponding level and the next higher level. For example, if the contour levels are 20, 30, 40,..., etc., then the color corresponding to level 20 is used to fill in the space between the level 20 contour and the level 30 contour.

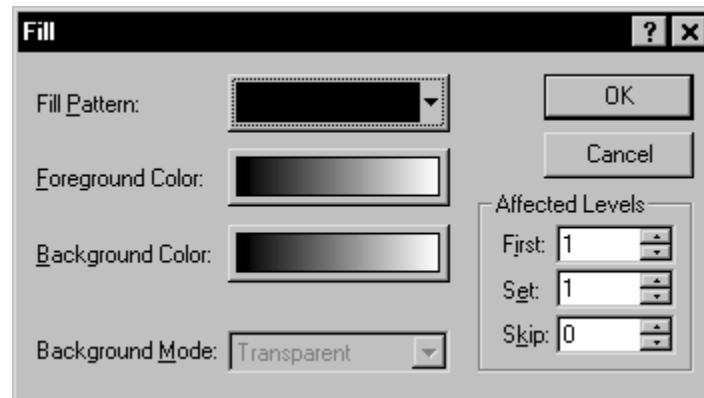
To change color fill:

1. Double-click the contour map and to open the contour map properties dialog.
2. Click the *Fill Contours* check box on the **General** page.



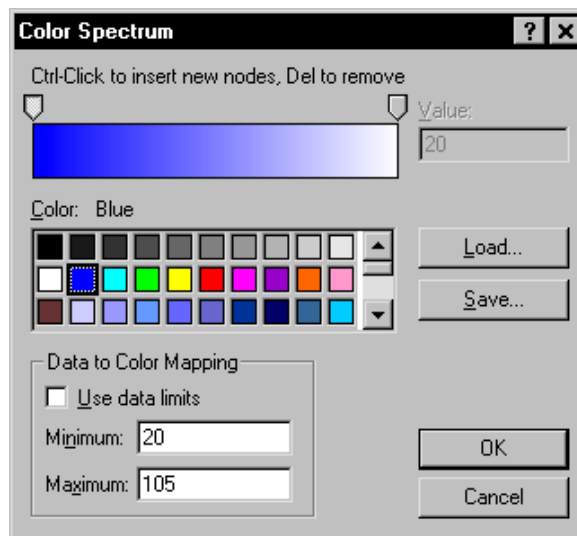
*Make sure the Fill Contours box is checked on the **General** page to add fill between the contour lines.*

3. On the **Levels** page, click the *Fill* button to open the **Fill** dialog.



*You can use the **Fill** dialog to set the contour level foreground and background fill.*

4. Click the *Foreground Color* button to open the **Color Spectrum** dialog. This dialog allows you to select colors to assign to specific Z values. Click on the left anchor point button above the spectrum, click on blue in the color palette, and then click OK. The *Foreground Color* button is now displayed as a gradation from blue to white in the **Fill** dialog.
5. Click OK and the fill colors on the **Levels** page are updated to reflect the change.
6. Click OK and the contour map is redrawn with color fill between the contours.




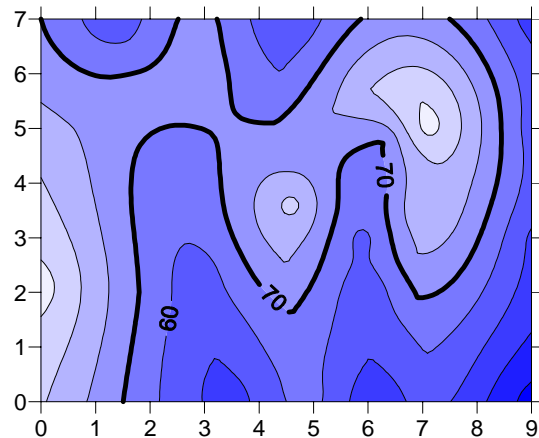
To change the color spectrum, click the left anchor button and then click blue in the color palette.

Add, Delete, and Move Contour Labels

Contour label locations can be changed on an individual basis. Labels can be added, deleted, or moved.

To add, delete, and move contour labels:

1. Right-click on the contour map and choose the **Edit Contour Labels** option. You can also edit labels of a selected contour map using the **Map | Contour Map | Edit Labels** command. The pointer changes to a black arrowhead  to indicate that you are in edit mode.
2. To delete a label, click on the label and press the DELETE key on the keyboard. For example, click on a 70 label and then click the DELETE key on your keyboard.
3. To add a label, press and hold the CTRL key on the keyboard and click on the location on the contour line where you want the new label located. Add a 60 contour label to the lower left portion of the map.
4. To move a contour label, click on the label, hold down the left mouse button, and drag the label. Move the 70 contour label on the right portion of the map to the north.
5. To exit the **Edit Contour Labels** mode, press the ESC key.



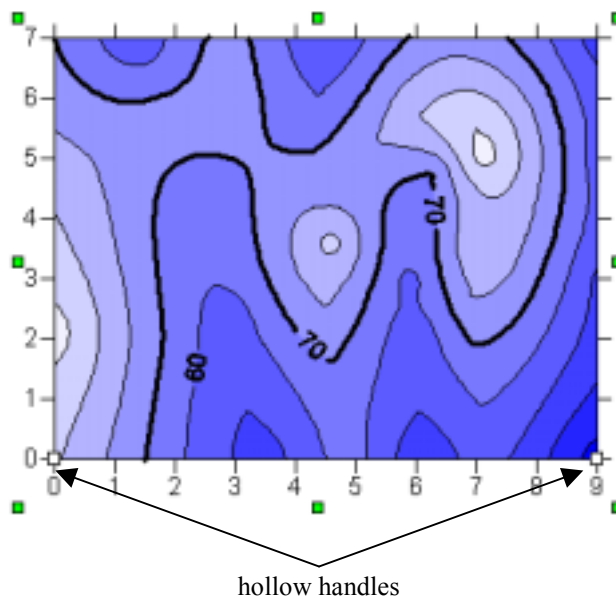
Contour labels can be moved, added, or deleted.

Modifying an Axis

Every contour map is created with four axes: the bottom, right, top, and left axes. You can control the display of each axis independently of the other axes on the map. In this example, we will change the contour spacing and add an axis label.

To modify an axis:

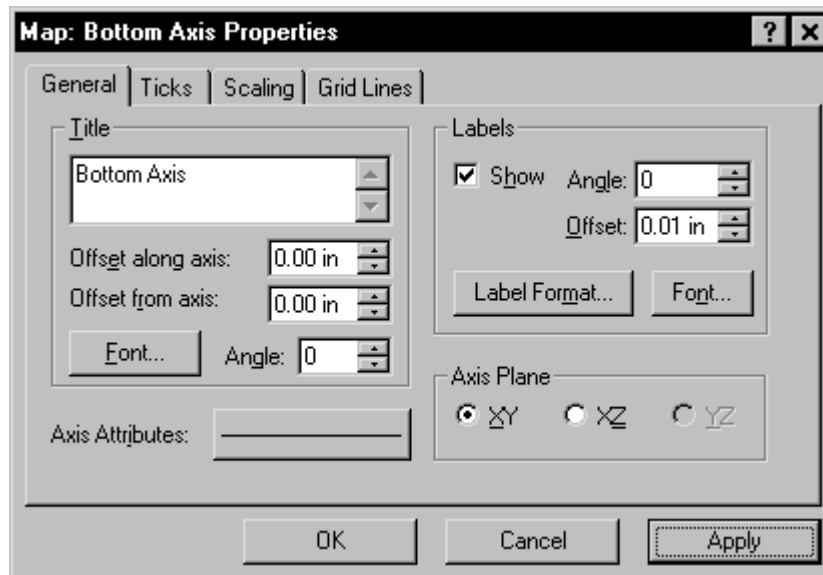
1. Move the pointer over one of the axis tick labels on the bottom X axis and left-click the mouse. In the status bar at the bottom of the plot window, the words "Map: Bottom Axis" are displayed. This indicates that you have selected the bottom axis of the contour map. Additionally, hollow handles appear at each end of the axis, and solid green handles appear surrounding the entire map. This indicates that the axis is a "sub-object" of the entire map.



Eight solid green handles appear around the selected map, and hollow handles appear at the ends of the selected sub-object. In this case, the bottom axis is the sub-object.

2. Double-click on the bottom axis to display the bottom axis properties dialog.

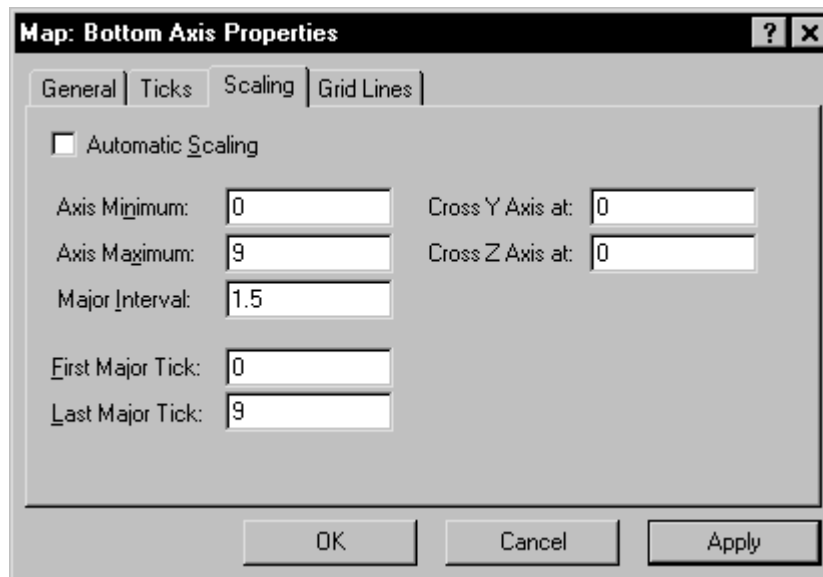
3. In the **Title** box on the **General** page, type "Bottom Axis" (without quotes) and then click the **Apply** button. This places a title on the selected axis.



If you would like to edit other axes, just click on the axis. You do not need to close the dialog before changing your selection.

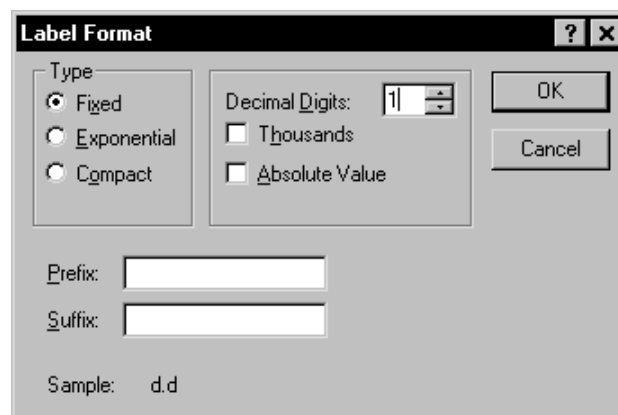
4. If you cannot see the axis title, select **View | Zoom | Selected**. Notice that you do not have to close the properties dialog to select menu commands, toolbar buttons, or objects in the plot window.

5. Click on the **Scaling** page to display the axis scaling options. In the *Major Interval* box, type the value 1.5 and then click the *Apply* button. This changes the spacing between major ticks along the selected axis.



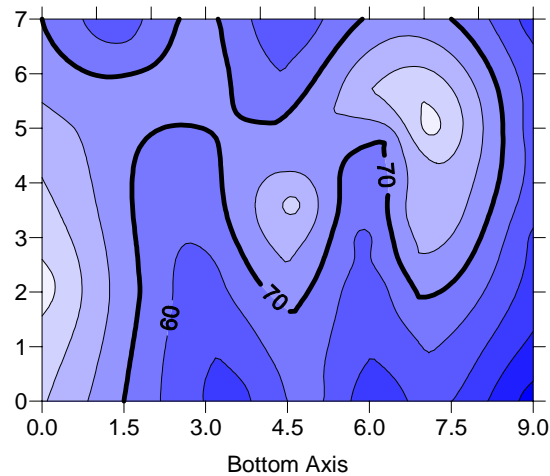
Enter 1.5 for the Major Interval to space the ticks 1.5 units apart.

6. Click on the **General** page and then click the *Label Format* button to open the **Label Format** dialog.
7. In the **Label Format** dialog, select the *Fixed* option in the *Type* group. Click on the down arrow on the *Decimal Digits* box and change the value to 1. This indicates that only one digit follows the decimal point for the axis tick labels.
8. Click OK in the **Label Format** dialog to return to the axis properties dialog.



*Set the label format to Fixed with one Decimal Digit in the **Label Format** dialog.*

- Click OK in the axis properties dialog and the map is redrawn. The axis tick spacing and labels are changed, and the axis title is placed below the map.




You can use the axis properties dialog to change the tick mark and axis title properties.

Saving a Map

When you have completed the map or drawing in the plot window, you can save the map to a **Surfer** file [.SRF] containing all the information necessary to reproduce the map. When you save a map, all the scaling, formatting, and parameters for the map are preserved in the file.

To save a map:

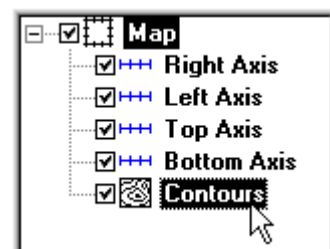
- Choose the **File | Save** command, or click the  button. The **Save As** dialog is displayed because the map has not been previously saved.
- In the *File name* box, type TUTORWS.
- Click *Save* and the file is saved to the current directory with an [.SRF] extension. The saved map remains open and the title bar changes reflecting the changed name.

Exporting 3D Contours

When you have completed a contour map in the plot window, you can export the contour lines with associated Z values to an AutoCAD DXF file.

To export contour lines:

1. Select the map by clicking on the map in the plot window or by clicking on the word "Contours" in the **Object Manager**.
2. Choose **Map | Contour Map | Export Contours**.
3. In the **Save As** dialog, type TUTORWS into the *File name* box.
4. Click *Save* and the file is exported to the current directory. This creates a file titled TUTORWS.DXF that can be opened in any program with an AutoCAD DXF import option.





*You can select the map by clicking on it in the **Object Manager** (above) or by clicking on the map in the plot window. If you do not see the **Object Manager**, click **View | Object Manager** to open it.*

Lesson 4 - Creating a Wireframe

Wireframes are three-dimensional representations of a grid file. Wireframes are block diagrams generated by drawing lines representing the grid X and Y lines (the grid columns and rows). At each intersection of a column and row (i.e. at each grid node), the height of the surface is proportional to the grid Z value at that point. The number of columns and rows in the grid file determines the number of X and Y lines drawn on the wireframe.

Creating a New Wireframe


To create a new wireframe:

1. Select the **File | New** command, or click the  button.
2. Select *Plot Document*, in the **New** dialog and click OK. A new empty plot window is displayed.
3. Select the **Map | Wireframe** command or click the  button.
4. Choose the grid file TUTORWS.GRD from the list of files in the **Open Grid** dialog, click *Open*, and the map is created. (TUTORWS.GRD, created in *Lesson 2 - Creating a Grid File*, is located in **Surfer's** SAMPLES folder.)

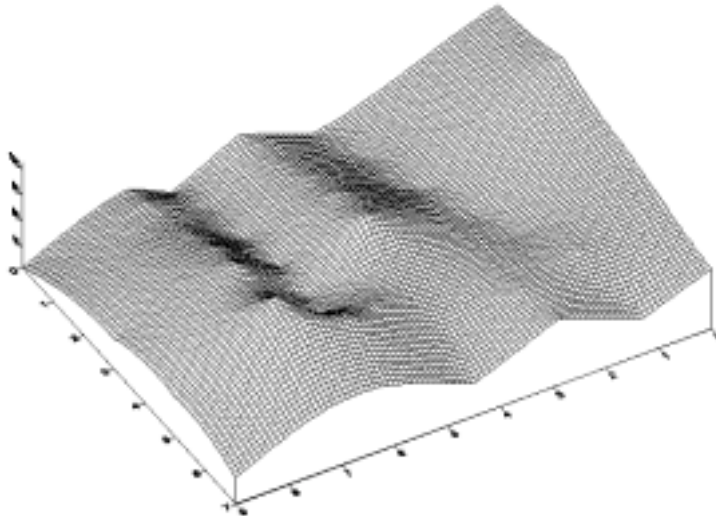
Changing the Orientation of a Wireframe

Although this example uses a wireframe to illustrate rotation and tilt, you can perform this operation on most map types in **Surfer**.

To change rotation and tilt:

1. Click on the map once to select it.
2. Click the **Map | Trackball** command or click the  button.
3. Click the left mouse button and hold it while moving the mouse to the left and right. This changes the rotation of the wireframe. The rotation is listed in the status bar.
4. Click the left mouse button and hold it while moving the mouse up and down. This changes the tilt of the map. The tilt is listed in the status bar.
5. Once you have rotated and tilted the map, click the ESC key on your keyboard to end trackball mode.

Changing the rotation, tilt, field of view, and projection can also be accomplished by double clicking the wireframe and using the options on the **View** page.



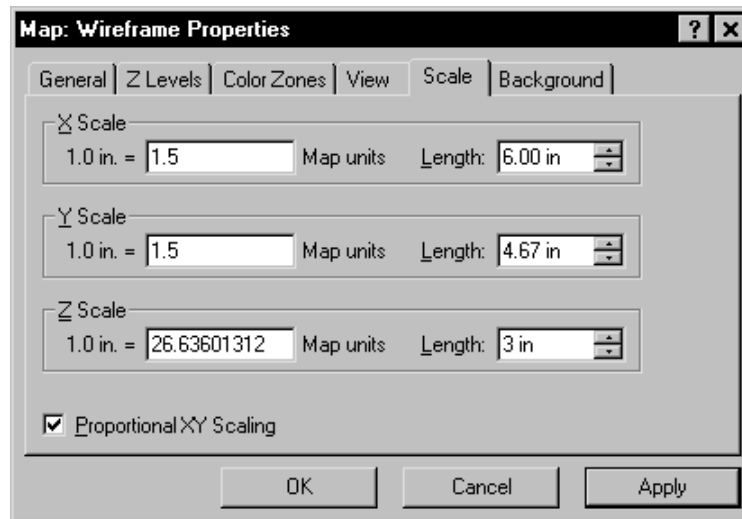
This is the wireframe at a new rotation and tilt.

Changing the Scaling of a Wireframe

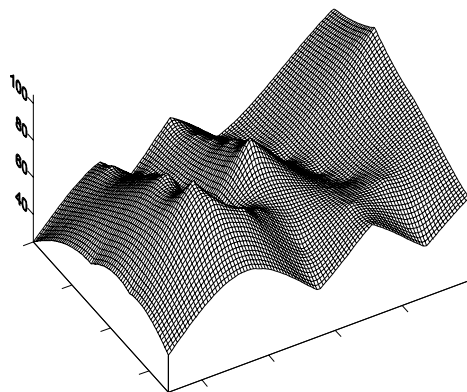
You can change the scaling of wireframes and surfaces in three dimensions. In this example, the Z scaling is changed, and the wireframe becomes somewhat exaggerated in the Z dimension.

To change the Z scaling:

1. Double-click the wireframe to open the map properties.
2. Click the **Scale** page. You can set scaling by entering the number of map units per inch or by entering the length for the map in page coordinates.
3. In the *Length* box of the *Z Scale* group, change the value to 3 or by scrolling to 3 or by highlighting the existing value and typing 3. The *Map units* value changes to reflect the change you made.
4. Click OK and the wireframe is redrawn exaggerated in the Z dimension.



On the Scale page, you can change the Z scaling on wireframes and surfaces.



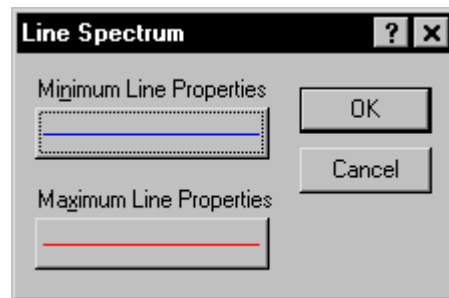
The Z scale on the wireframe is changed Z dimension is exaggerated.

Adding Color Zones to a Wireframe

You can change the line colors of any wireframe by applying color zones. In this example, we will change color zones gradationally and individually.

To add a color zone to a wireframe:


1. Double-click on the wireframe, to display the wireframe properties dialog.
2. Click on the **Color Zones** page.
3. Click on the *Line* button to display the **Line Spectrum** dialog.
4. Click on the *Minimum Line Properties* button to open the **Line Properties** dialog. From here, you can select the line color, style, or thickness.

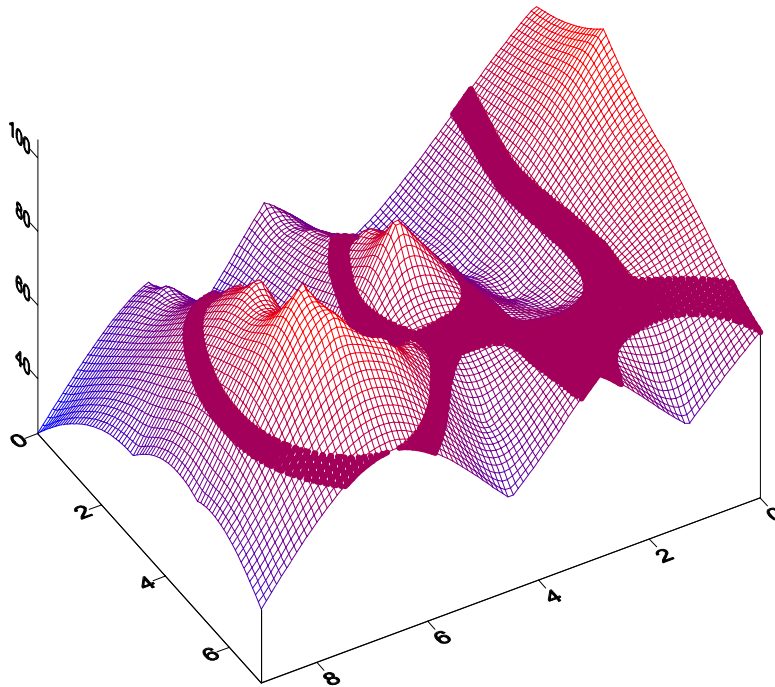


*Click the Line button on the **Color Zones** page to open the **Line Spectrum** dialog.*

5. Click on the *Color* button and select blue.
6. Click OK to return to the **Line Spectrum** dialog.
7. Click on the *Maximum Line Properties* button and change the line color to red using the steps above.
8. Click OK in the **Line Spectrum** dialog to return to the **Color Zones** page.
9. In the wireframe properties dialog, check the *Apply Zones to Lines of Constant X* and *Y* boxes by clicking in them.
10. Click *Apply* and the wireframe is displayed with gradational colors varying by the Z variable. (Leave the dialog open.)

To change the properties of an individual Z value:

1. On the **Color Zones** page, double-click the line sample for the contour level at $Z = 70$.
2. You can select the line color, style, or width for the selected line in the **Line Properties** dialog. In the *Width* box, click the up arrow and change the width value to 0.030 in.
3. Click OK in the **Line Properties** dialog and the **Color Zones** page is updated to reflect the change.
4. Click OK in the wireframe properties dialog and the map is redrawn. The color zone at $Z = 70$ is drawn with a thicker line, and is emphasized on the map. (If you need to zoom in on the map, click the  button and then click on the map. Click the ESC key on the keyboard after you are finished zooming in.)



Double-click on the 70 level line sample to emphasize the $Z = 70$ contour level.


Lesson 5 - Posting Data Points and Working with Overlays

Post maps are created by placing symbols representing data points on a map. Posting data points on a map can be useful in determining the distribution of data points, as well as placing data or text information at specific points on the map. Data files contain the XY coordinates used to position the points on the map. Data files can also contain the labels associated with each point.

Creating Post Maps

When a new post map is created, it is independent of any other map in the current plot window. When the two maps are displayed, notice that two sets of axes are also displayed, one set for each map. When you use the **Map | Overlay Maps** command, the two maps are combined into a single composite map with one set of axes.

To create a post map:

1. Choose **File | Open** to display the **Open** dialog. Choose TUTORIAL.SRF in the SAMPLES directory and then click the *Open* button. TUTORIAL.SRF opens and contains a contour map.
2. Select the **Map | Post Map | New Post Map** command or click the  button.
3. Choose TUTORWS.DAT from the list of files in the **Open** dialog. (TUTORWS.DAT is located in **Surfer's** SAMPLES folder.)
4. Click *Open* and the post map is created using the default properties.

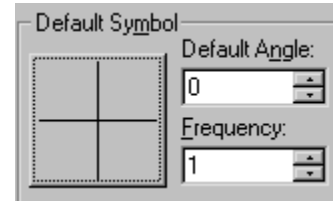
The contour map and post map are two separate maps on the page. If you look closely at the X axis, you will notice the two sets of axis tick labels for the two maps. Also, there are two map frames and axes sets listed in the **Object Manager**. We will line up the maps with the **Overlay** command after changing the post map properties.

Changing the Post Map Properties

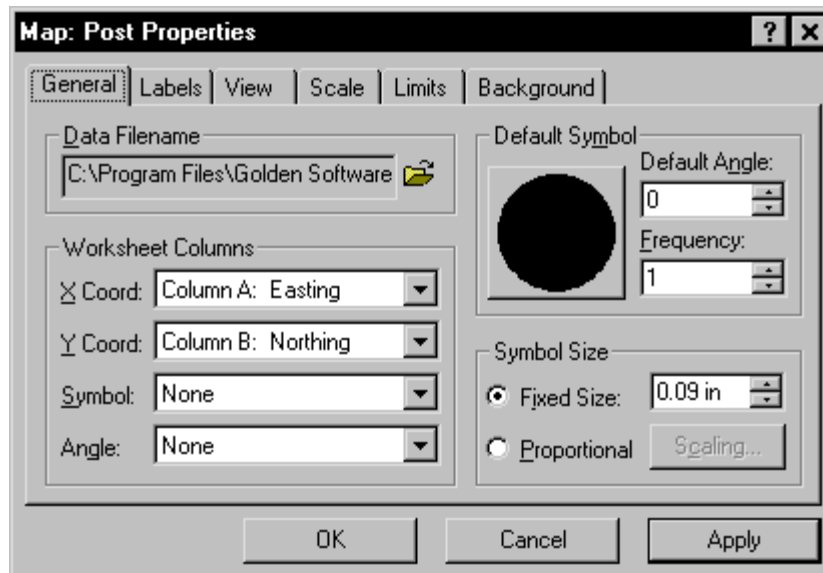
To change the post map properties:

1. Open the **Object Manager** if it is not already open. The **Object Manager** is open if there is a check mark displayed next to the words **Object Manager** in the **View** menu. If there is no check mark, click on the command once.
2. Right click on the word "Post" in the **Object Manager**.
3. Select **Properties** from the context menu.

4. In the post map properties dialog **General** page, click the symbol button to open the **Symbol Properties** dialog.
5. Choose the filled circle symbol from the palette and click OK. The selected symbol appears in the symbol button.
6. Click the *Apply* button and the symbol appears at the posted data points on the map.
7. In the *Fixed Size* box (*Symbol Size* group), specify a size of 0.09 in.
8. Click OK and the post map is drawn with the new symbol size.



*The symbol button is located in the Default Symbol group on the **General** page.*

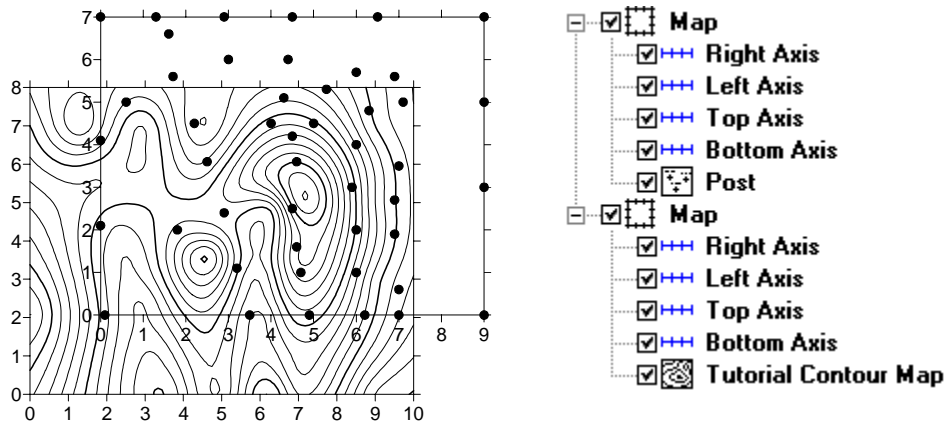


*You can change the symbol size to 0.09 inches on the **General** page.*

Overlaying Maps

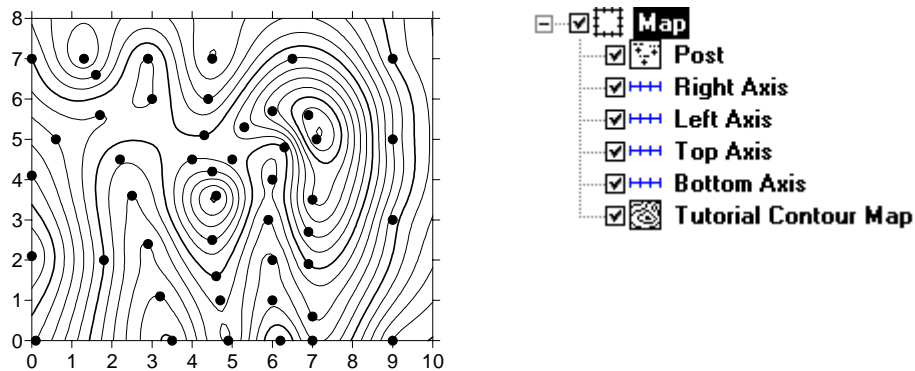
To overlay maps:

1. To see the two separate maps, place the mouse pointer in the center of the maps and click. Press and hold the left mouse button and move the pointer slightly in any direction. Release the button and the two maps are offset.



*The two maps are offset to show that the contour map and post map are separate maps. They also appear as two separate maps in the **Object Manager** as shown on the right.*

2. Click **Edit | Select All** to select both maps.
3. Choose the **Map | Overlay Maps** command to combine the two maps into a single composite map.



*The contour and post maps are combined into a single composite map after using **Overlay Maps**. Notice that the **Object Manager** displays one "Map" containing a post map, four axes, and the contour map.*

Selecting an Overlay and Assigning an Object ID

After creating a composite map, you can still modify the individual overlays in the map.

Selecting Maps

An individual overlay can be selected in the composite map by clicking on the overlay in the plot window or clicking on the overlay in the **Object Manager**.

The easiest way to select an overlay in a composite map is to click on the overlay name in the **Object Manager**. However, you can also select the overlay in the plot window with the mouse. Whenever two or more objects occupy the same position in the plot window, use the CTRL key and the left mouse button to select the desired object. The CTRL key allows you to cycle through the selection of overlapping objects. For example, if you want to select a text block behind a rectangle, hold down the CTRL key and click until the text is selected. You can use the status bar to help you to determine which object is selected.

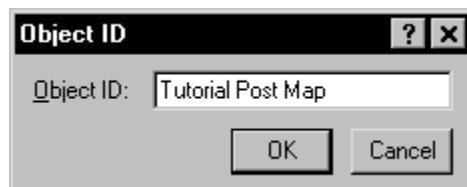
Renaming the Post Map

To select an overlay and assign an ID:

1. Make sure the **Object Manager** is open. If the **Object Manager** is not open, click **View | Object Manager**.
2. Click the overlay name in the **Object Manager**. In this case, click the word "Post." The status bar should now report "Map: Post."
3. Choose **Edit | Object ID**.
4. In the **Object ID** dialog, type the name "Tutorial Post Map" and click OK. The status bar, **Object Manager**, and properties dialog title reflect the name change.



*Click the word Post in the **Object Manager** to select the post map.*



*You can change the name of the selected object using **Edit | Object ID**.*

Chapter 2 - Tutorial

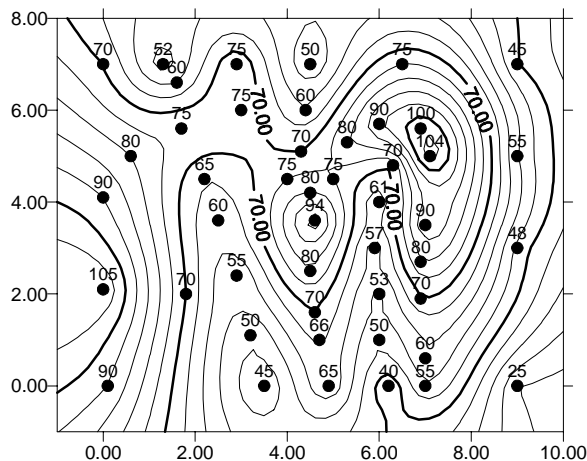
If you double-click on Tutorial Post Map in the **Object Manager**, notice that the properties dialog title changes to **Map: Tutorial Post Map Properties**. When you rename an object in **Surfer 8**, the object's properties dialog reflects the name change making it easier for you to keep track of the object you are editing. For example, if you have eight post maps in the plot window, it is beneficial to change the post map names to something meaningful to save time when trying to edit them. This is especially important because the properties dialog can stay open when changing selections.

Adding Labels to the Post Map

You can add labels to the data points on post maps. The post map can be selected by a few different methods, though only the **Object Manager** method is discussed here.

To add labels:

1. Right-click on "Tutorial Post Map" in the **Object Manager** and choose **Properties**.
2. Click on the **Labels** page. In the *Worksheet Column for Labels* group, click the drop-down arrow and a list of columns in TUTORWS.DAT is displayed.
3. Select *Column C: Elevation* from the list.
4. Click the *Format* button to open the **Label Format** dialog.
5. Change the *Type* to *Fixed* and the *Decimal Digits* value to zero.
6. Click OK to return to the post map properties dialog.
7. Click OK and the overlay is redrawn with labels on each of the data points.



You can add labels to post maps through the post map properties dialog.

Lesson 6 - Introducing Surfaces

Surfaces are new to **Surfer 8**. They are similar to wireframes, except that surfaces are three-dimensional shaded renderings of a grid file. Surfaces provide an impressive visual interpretation of data. Surfaces can be overlaid with other surfaces, so that the surfaces will intersect with each other. Surfaces can also have overlays of other map types, excluding wireframes.





This is a surface of the Telluride, Colorado USGS SDTS file.

You can control the color, lighting, overlay blending, and wire mesh grid of a surface.

Creating a Surface

To create a surface:

1. Select the **File | New** command, or click the  button.
2. Select *Plot Document*, in the **New** dialog and click OK. A new empty plot window is displayed.
3. Select the **Map | Surface** command or click the  button.
4. Choose the grid file TUTORWS.GRD from the list of files in the **Open Grid** dialog, click *Open*, and the surface is created using the default settings. (TUTORWS.GRD, created in *Lesson 2 - Creating a Grid File*, is located in **Surfer's** SAMPLES folder.)


Adding an Overlay

You can overlay base, contour, post, image, shaded relief, and vector maps on surfaces. All overlays, except other surfaces, are converted into a type of bitmap known as a texture map. This texture map is then applied to the surface by stretching it and shrinking it as necessary. When these maps are overlaid, you have a choice on how to treat the texture map. You can use the colors from overlays only, from the surface only, or blend colors from the overlays and surface. For example, you could create a color filled contour map, overlay the contour map and surface, and then use the colors from the contour map only.

Chapter 2 - Tutorial

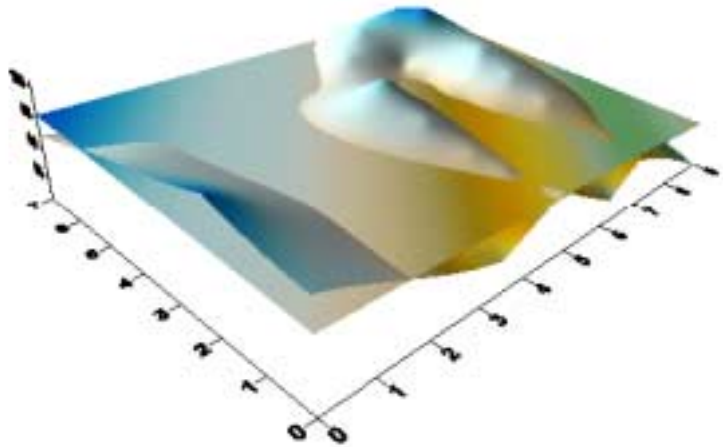
When multiple surfaces of differing elevations are overlaid, the surfaces can intersect and overlap each other. If the surfaces are adjacent to each other in the X or Y direction, the surfaces are drawn side-by-side after using **Map | Overlay Maps**. In this example, we will overlay a plane with the surface you just created.

First, create the surface plane:

1. Select the **Map | Surface** command or click the  button.
2. In the **Open Grid** dialog, open **Surfer's** SAMPLES folder and select TUTORPL.GRD.
3. Click *Open*, and the surface is created using the default settings.
4. Right click on the upper 3D Surface in the **Object Manager**, choose **Object ID** and add the text TUTORPL.GRD to the end of the 3D Surface label to make it easier to distinguish the two surfaces in the **Object Manager** list.

Next, overlay the surfaces:

1. Click **Edit | Select All** to select both surfaces.
2. Click **Map | Overlay Maps** to overlay the surfaces.



You can overlay two or more surfaces. Depending on each surface's XYZ ranges, the surfaces may overlap or intersect each other.

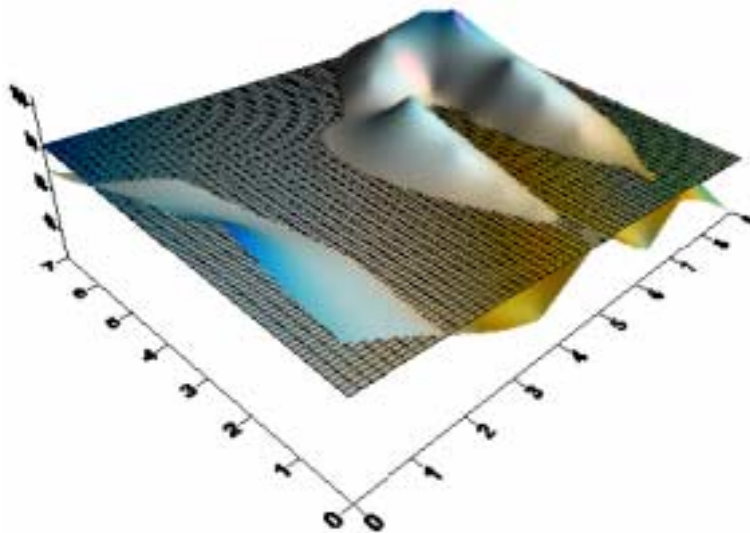
Adding a Mesh

Mesh lines can be applied to surfaces.

To add a mesh:

1. Double-click on the surface to open the surface properties.
2. Click the **Mesh** page.
3. Check the *X* and *Y* boxes.
4. Change the *Frequency* to 5 for the *X* and *Y* lines.
5. Click the OK or *Apply* button to add a mesh to the selected surface.

Notice that the mesh is applied to the selected surface within the composite map, not to both surfaces. With all map types, you can only change the map-specific properties of one map at a time. Properties that apply to all overlays in the composite map include **View**, **Scale**, **Limits**, and **Background**.





The mesh is applied to the selected overlay in the composite map.

Changing Colors

Changing color schemes on surfaces is similar to changing colors on other map types such as image maps or contour maps. The **Color Spectrum** dialog is used to load previously defined color schemes, and it is also used to create your own color schemes. The **Color Spectrum** dialog is discussed in greater detail in *Chapter 16, Adding Color to Maps*.

Before experimenting with color, let's create a new surface map in a new plot window:

1. Click **File | New** or click the  button.
2. In the **New** dialog, select *Plot Document* and then click OK.
3. Select the **Map | Surface** command or click the  button.
4. In the **Open Grid** dialog, open **Surfer's** SAMPLES folder and select any grid file [.GRD]. HELENS2.GRD is a good choice for experimenting with color.
5. Click *Open*, and the surface is created using the default settings.

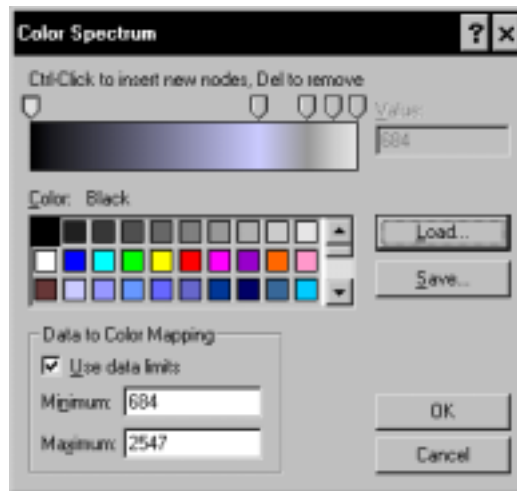
To change the surface material color:

1. Double-click on the surface to open the surface properties.
2. There is a *Material Color* group on the **General** page. Click the *Upper* button.




Click the Upper button in the Material Color group to change the surface material color.

- Click the *Load* button in the **Color Spectrum** dialog.



Click the Load button to use a different color spectrum file [.CLR] for the surface material color.

- The color schemes are stored in color spectrum files containing a [.CLR] extension. By default, **Surfer** opens in the SAMPLES folder, which contains many predefined color spectrums. (Browse to the SAMPLES folder if it is not open.) Click on one of the [.CLR] files to select it and then click the *Open* button.
- Notice that the colors and anchor node  positions have changed in the **Color Spectrum** dialog. Click OK in the **Color Spectrum** dialog to return to the surface map properties dialog.
- Click the *Apply* button in the surface map properties dialog to see your color changes. Drag the surface map properties off to the side if you cannot see the map.

You can continue to experiment with the colors by clicking the *Upper* button and loading other color spectrum files.

Need More Information?

If you find you still have questions after you have completed the tutorial, you should consider reviewing the material in the rest of the User's Guide or accessing **Surfer's** extensive online help. Usually, the answers to your questions are found in these locations. However, if you find you still have questions, do not hesitate to contact Golden Software's technical support. We are happy to answer your questions before they become problems.

Chapter 3

Data Files and the Worksheet

Introduction to Surfer Data and Worksheet Windows

Data Files

Data files contain the raw information used to create a grid file, perform residual calculations, or produce post maps. Each record in a data file occupies a single row and is comprised of at least two values (XY) for post maps and at least three values for gridding (XYZ). The X, Y, and Z values are each placed in separate columns. X and Y coordinates define the position of the point on the map, and the Z value defines the value assigned to the specific XY location. Common examples of XY coordinates include longitude-latitude, Easting and Northing, or UTM (Universal Transverse Mercator) coordinates. Z data might be topographic elevation, water depth, chemical concentrations, temperature, or any other quantity amenable to mapping.

Data files can be created in the **Surfer** worksheet, an ASCII editor, or any program that can produce files in one of the file formats listed in this chapter. Regardless of the program used to create your data files, you must save the file on disk prior to performing any **Surfer** operation requiring a data file, including the gridding operation. **Surfer** reads data only from a data file in one of the recognized formats listed in the *Opening Data Files* section of this chapter.

When one of the commands calling for a data file is chosen, the data are loaded into an internal worksheet. It is not necessary to place the data in a worksheet window to perform any of the operations requiring data files, such as the gridding operation. If you want to view or alter the data in a data file, you can use the **File | Open** command to gain access to the worksheet data.

XYZ Data Files

XYZ data files contain the raw data **Surfer** interprets to produce a grid file. Before you create a grid file in **Surfer**, you must create an XYZ data file. XYZ data files must be organized in column and row format. By default, **Surfer** expects the X data to be contained in Column A, the Y data in Column B, and the Z data in Column C. However, the data can be placed in any order in any column.

Chapter 3 - Data Files and the Worksheet

Portions of two simple data files are shown below. The order of the data in the file is not important. These examples contain descriptive headers in Row 1 of each column. Such information is helpful but not required by **Surfer** to create a grid file. When text appears in Row 1 of a column, this text appears in list boxes in various **Surfer** dialogs as column titles. If a number resides in Row 1, it is not incorporated into the dialogs, and instead, the column heading (such as Column B) is displayed.

	A	B	C
1	0.1	0	90
2	3.5	0	45
3	4.9	0	65
4	6.2	0	40
5	7	0	55
6	9	0	25
7	9	5	55
8	9	3	48

This is a simple XYZ data file.

	A	B	C
1	Longitude	Latitude	Elevation
2	-109	39.205	1464
3	-108.965	39.337	1524
4	-108.93	39.389	1385
5	-108.895	39.526	1583
6	-108.86	39.588	1445
7	-108.825	39.795	1371
8	-108.79	40.003	1371

This is another example of an XYZ data file with header information in Row 1 of each column in the data file.

Rows with non-numeric entries (empty cells or text) in any of the X, Y, or Z columns are excluded when performing various tasks such as gridding and transforming data in the worksheet. If there is no Z information for a particular XY location, you can leave the Z column blank in that row. In the example shown here, there are two data records without Z values. These records are not considered during the gridding operation.

Data files can contain up to one billion columns. Since you can specify the columns to be gridded, your X, Y, and Z values can occupy any three columns. This allows you to have columns containing other information particular to each point. The data file can contain several Z columns, so you can produce several contour maps using the same XY coordinates. For example, you might have concentrations of different contaminants at each sample location. All the contaminant concentration data can be placed in the same data file.

	A	B	C
1	X Data	Y Data	Z Data
2	0.1	0	90
3	9	3	48
4	1.3	7	
5	4.7	1	66
6	1.7	5.6	75
7	6	1	
8	2.5	3.6	60

Blank Z column cells are ignored when gridding a data file.

	A	B	C	D	E	F	G	H
1	X Coord	Y Coord	SS-A	SH-A	SLT-B	SH-B	SS-B	SA-C
2	4345	14005	1432	1478	1503	1593	1627	1645
3	4355	14015	1437	1483	1501	1589	1625	1640
4	4365	14025	1445	1479	1498	1592	1637	1638
5	4375	14035	1434	1466	1515	1586	1626	1635
6	4385	14045	1435	1476	1510	1597	1616	1637
7	4389	14055	1448	1475	1512	1601	1603	1638

This is an example of an XYZ data file containing several columns of Z data. You could use this file to create several different grid files, where each uses the same XY coordinates, but different Z data.

Data files may contain data in addition to the X, Y coordinates, and the Z values. For example, when posting data with the **Map | Post Map** command, additional columns can be used to specify the symbol, the rotation angle, labels, etc. The following is an example of such a data file. Columns A, B, and C contain the XYZ data used to produce a contour map of depth to the water table. Columns D, E, and F contain information used to create an overlaying post map.

	A	B	C	D	E	F
1	Easting	Northing	Depth	Well Name	Label Angle	Symbol Number
2	14326	7753	576	TC-3	30	104
3	14375	7659	489	TC-7	30	104
4	14426	7575	503	WC-5	30	110
5	14537	7283	577	TC-13	45	104
6	13498	7444	589	WC-9	50	110
7	17521	7366	501	TC-8	35	104

A data file used to create a post map or a classed post map can contain several columns of data. Each column can have a different effect on the posted data points.

Data as Numbers or Text

Worksheet data is in one of two forms: numbers or text. Grid file creation, statistics, post maps, and other operations require data as numbers. Text data (even if it contains numeric digits) can be used for labels in **Surfer**, but it cannot be used to create grids or in any operation requiring numbers.

Numbers can consist of digits (0 - 9), decimal points (.), the letters "e," "d," "E," or "D" (indicating an exponent), and the plus (+) or minus (-) sign. If you type any characters other than these when entering a number (or type any of the special characters more than once), **Surfer** automatically converts the cell entry to text. For example, if your longitude data appears as 104.5 W in a worksheet cell, it is interpreted as text and cannot be used to grid data.

Chapter 3 - Data Files and the Worksheet

You can also convert numeric data to text by typing a single quotation mark (') in front of the number. **Surfer** does not place the single quotation in the worksheet cell, however.

By default, numeric data is right justified in a cell, and text is left justified. However, cell entries, whether numeric or text, can also be justified by specifying the desired alignment using the **Alignment** page through **Format | Cell Properties**.

	B25		'1.6
	A	B	C
25	4.6	1.6	70
26	4.5	2.5	80
27	4.6	3.6	95
28	4.5	4.2	80
29	4.3	5.1	70
30	4.4	6	60
31	5.3	5.3	78
32	6	5.7	88

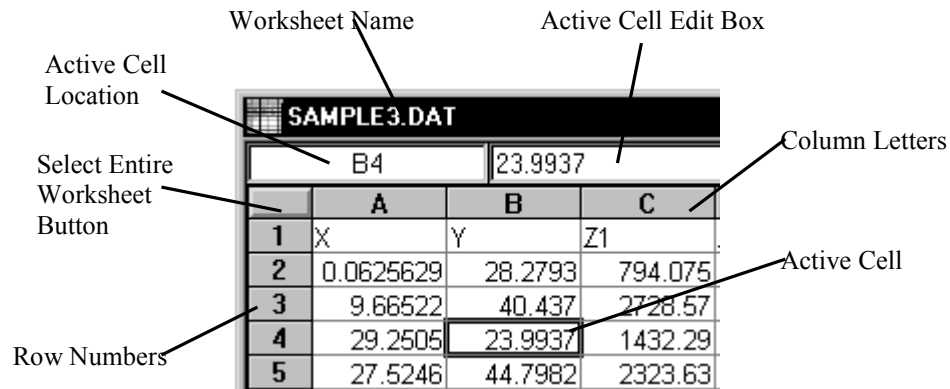
Notice that Column B is left aligned. This means the numbers are formatted as text. When a cell is highlighted, an apostrophe appears in the active cell edit box, also indicating that the number is formatted as text.

Worksheet

Worksheet windows are a view of the data file and are designed to display, edit, enter, and save data. The worksheet windows have several useful and powerful editing, transformation, and statistical operations available. Several import and export options are available for opening data files from other spreadsheet programs.

The Worksheet Window

The components of the worksheet window are discussed below.



The components of a worksheet window shown above are described in the following table.

Column Letters	The letter that identifies a column of the worksheet.
Row Numbers	The number that identifies a row of the worksheet.
Active Cell	The cell highlighted with a bold outline. The active cell receives data input (numeric values or text strings) from the keyboard. Only one cell is active at a time.
Active Cell Location	The location of the active cell, specified by column letter and row number.
Active Cell Edit Box	The box displaying the data or text contained in the active cell. Data typed into an empty cell appears in both the edit box and the active cell.
Worksheet Name	The name of the data file displayed in the worksheet or the worksheet number prior to saving.
Select Entire Worksheet Button	The button used to select all cells in the worksheet.

Working with Data

There are three ways to enter data into the worksheet. You can use **File | Open** to open an existing data file, type data directly into the worksheet, or copy the data from another application and paste it into the worksheet.

There are two basic modes in the worksheet. Normal mode is when the active cell can be moved throughout the worksheet, and edit mode allows the contents of a single cell to be edited in the active cell edit box. Only one mode may be active at a given time. ESC, ENTER, or clicking on another cell can be used to exit edit mode and return to normal mode.

Entering Data into a Cell

The **Active Cell** marks the position of the cell that can be edited. A heavy line surrounding the cell indicates the active cell, and its location is given in the **Active Cell Location** box. The worksheet cell is selected as the active cell by clicking on the cell with the mouse, by using the PAGE UP, PAGE DOWN, TAB, SHIFT+TAB, HOME, END, and ARROW keys.

To enter new data and delete old data, highlight the contents in the **Active Cell Edit Box** and begin typing. Edit mode is entered automatically and the old cell contents are deleted, if there is anything in the cell. "Edit mode" means that the cell contents can be changed. Press ENTER, an ARROW key, TAB, SHIFT+TAB, PAGE UP or PAGE DOWN keys to permanently record data in the cell.

Cells can also be edited by pressing the F2 key or by highlighting the contents of the **Active Cell Edit Box**. Use the HOME, END, BACKSPACE, DEL, and ARROW keys to move the cursor within the cell. Press the ENTER, TAB, or SHIFT+TAB keys to permanently record information in the cell. Press the ESC key to cancel the changes and restore the original data.

Moving the Active Cell

You can designate any worksheet cell as the active cell by clicking on it with the mouse. The active cell can also be repositioned by using keyboard commands.

ARROW keys (Up, Down, Left, Right) move the active cell to an adjacent cell.

PAGE UP/PAGE DOWN moves the active cell up or down by the number of rows visible in the window.

HOME moves the active cell to the first occupied cell in the current column. Pressing HOME again moves the active cell to the top row in the current column.

END moves the active cell to the last occupied row in the current column. Pressing END again moves the active cell to the bottom row of the worksheet.

ENTER moves the active cell down one row and ends edit mode.

TAB moves the active cell right one column and ends edit mode.

SHIFT+ENTER moves the active cell up one row and ends edit mode.

SHIFT+TAB moves the active cell left one column and ends edit mode.

CTRL+HOME moves the active cell to the top cell of the left most column in the worksheet (A1).

CTRL+END moves the active cell to the bottom occupied row of the last occupied column in the worksheet.

The CTRL+LEFT ARROW behavior depends on the position of the active cell. If the active cell is to the right of the last occupied column in the current row, it moves the active cell to the last occupied column in the current row. If the active cell is in or to the left of the last occupied column in the current row, but to the right of the first occupied column in the current row, it moves the active cell to the first occupied column in the current row. Otherwise, CTRL+LEFT ARROW moves the active cell to the first column in the current row.

The CTRL+RIGHT ARROW behavior depends on the position of the active cell. If the active cell is to the left of the first occupied column in the current row, it moves the active cell to the first occupied column in the current row. If the active cell is in or to the right of the first occupied column in the current row, but to the left of the last occupied column in the current row, it moves the active cell to the last occupied column. Otherwise, CTRL+RIGHT ARROW moves the active cell to the last column in the current row.

The CTRL+UP ARROW behavior depends on the position of the active cell. If the active cell is below the bottom occupied row in the current column, it moves the active cell to the bottom occupied row in the current column. If the active cell is below the top occupied row in the current

Chapter 3 - Data Files and the Worksheet

column, but in or above the bottom occupied row in the current column, it moves the active cell to the top occupied row in the current column. Otherwise, CTRL+UP ARROW moves the active cell to the first row in the current column.

The CTRL+DOWN ARROW behavior depends on the position of the active cell. If the active cell is above the top occupied row in the current column, it moves the active cell to the top occupied row in the current column. If the active cell is above the bottom occupied row in the current column, but below the top occupied row in the current column, it moves the active cell to the bottom occupied row in the current column. Otherwise, CTRL+ DOWN ARROW moves the active cell to the last row in the current column.

If a block of cells is selected, the ENTER, TAB, SHIFT+ENTER, and SHIFT+TAB keys move the active cell within a group of selected cells without canceling the selection.

Pasting Data

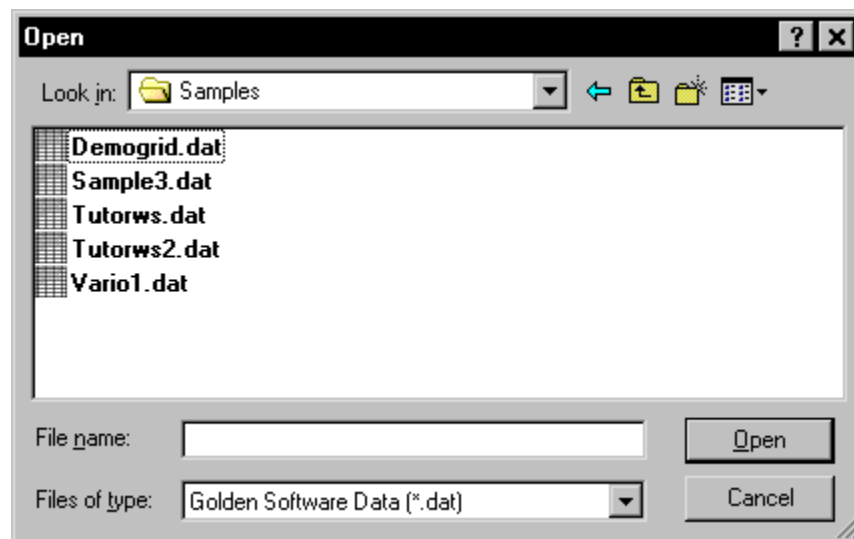
If data are copied to the clipboard from another software application, the contents of the clipboard can be pasted into the worksheet. If the source application is Microsoft Excel, some formatting information is retained. When pasting data into the worksheet, select a cell and use **Edit | Paste** (CTRL+V). Any data to the right or below the active cell is overwritten, so be sure to locate the active cell carefully. When data are copied to the clipboard, special formatting information is also copied. The **Edit | Paste Special** command determines the format in which the contents are pasted into the worksheet.

Opening Data Files

When you create a grid file or use another command that requires data, you do not need to open the data into the worksheet first. However, the worksheet is available if you would like to view or edit your data. In the worksheet, the **File | Open** command loads a data file into a new worksheet.

If the worksheet already contains data, additional data can be merged into the worksheet using the **File | Import** command. The contents of the new file are merged into the worksheet at the active cell so it is imperative that the cell be positioned at the edge of the existing data. Any cells in the existing worksheet that lie to the right and below the active cell will be overwritten with the contents of the merging file.

Multiple files can be opened at one time into the same worksheet with **File | Import** using the SHIFT or CONTROL keys while selecting files in the dialog.



*You can open several data file types from the **Open** dialog.*

Data File Formats

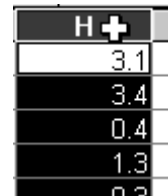
Data file formats are discussed in *Appendix C, File Formats*.

Editing the Data Files in the Worksheet

Formatting the worksheet, calculating statistics, and transforming data require the selection of one or more blocks of cells before performing the operation. A block of cells can be selected with the mouse by highlighting cells, row numbers, or column letters. Cells can also be selected with the keyboard. Selected cells are indicated by reverse video color.

Tips on Selecting Cells

- Clicking on the small box above the row labels and to the left of the column label bar selects the entire worksheet.
- To deselect all selected cells, click the left mouse button anywhere within the worksheet, press the ESC key, or move the active cell with an ARROW key or other movement key.
- To rapidly select a large block, first select one corner of the block, and then use the scroll bars to scroll to the opposite corner. Hold down the SHIFT key and click on the cell at the opposite corner.
- To select all cells in a column or row, click the column letter or row number. To select several adjacent columns or rows, press and hold the left mouse button and drag the pointer on the column letters or row numbers.
- While holding down the CTRL key, the active cell may be repositioned for selecting a new, discontinuous block.
- The CTRL key is used to select multiple blocks and the SHIFT key is used to resize the last selected block.
- Additional information on selecting cells with the mouse and keyboard is located in the online help (see *Selecting Cells*).



Click a column letter or row number to select an entire row or column.

Editing the Worksheet

The **Edit** menu contains several standard edit options such as **Undo**, **Redo**, **Cut**, **Copy**, and **Paste**. See *Chapter 21, Creating, Selecting, and Editing Objects* for more details on these commands. Some of the edit commands have functions specific to the worksheet and some are available through the context menus as well as the main menus. You can right-click on selected cells to open the context menu.

Paste Special

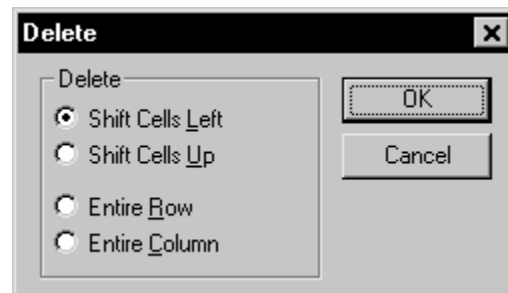
When data are copied to the clipboard, special formatting information is also copied. The **Paste Special** command determines the format in which the contents are pasted to the worksheet.

Clear

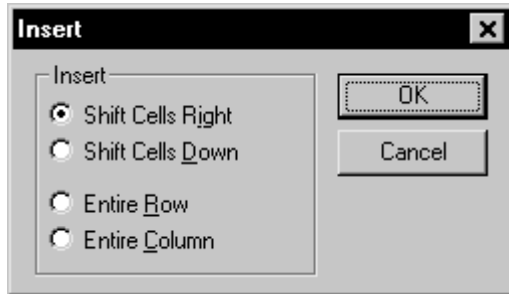
The **Clear** command removes data from selected worksheet cells. The cells are left empty when the data are removed. To shift the remaining data into the cleared cells, use the **Delete** command instead.

Delete

The **Delete** command deletes the selected worksheet cells and shifts the cells up or to the left to fill in the gap. After selecting **Delete**, click the *Shift Cells Up* or *Shift Cells Left* option button and then click OK. The selected cells are deleted and the contents of cells below or to the right are moved to fill the deleted block. Click *Entire Row* or *Entire Column* to delete the entire row or column that contains highlighted cells. To leave the selected cells empty when the data are removed, use the **Clear** command, press the DEL key, or use the **Cut** command.



*When using **Edit / Delete**, you can shift cells to the left or up to fill in the gap.*



*When using **Edit | Insert**, you can shift cells to the right or down to make room for the new cells.*

Insert

The **Insert** command inserts a single blank cell or a block of blank cells in the worksheet. Select a block of cells in the area in which you wish to insert cells and then select **Edit | Insert**, or right-click and choose **Insert** from the context menu. Click the *Shift Cells Down* or *Shift Cells Right* option button and then click OK. The blank cells are inserted and the original contents of the displaced cells are moved. Click *Entire Row* or *Entire Column* to insert an entire row or column in the area that contains highlighted cells.

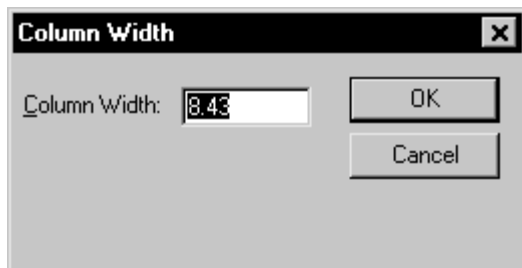
Modifying the Worksheet and Data Files

The worksheet can be formatted in various ways. Once the data are opened, imported, entered, or pasted into the worksheet they can be modified mathematically, formatted, and sorted. To save some of the formatting with the data file, choose Excel Spreadsheet [.XLS] when you save the file.


Column Width

You can change the column width of selected cells by typing a value into the **Column Width** box, or use the mouse to resize the column. The Excel Spreadsheet [.XLS] or Microsoft SYLK [.SLK] file format must be used to save the column width information within the file, since ASCII files such as [.CSV], [.TXT], [.DAT], [.BLN], and [.BNA] do not save format information.

To set column widths or to hide columns, select either the entire column or individual cells within the columns, and then select **Format | Column Width**. For information on selecting columns, see the *Tips on Selecting Cells* section in this chapter. Enter the width for the selected columns or cells in the **Column Width** dialog. Columns can range from zero to 512 characters wide. The value zero (0) hides the column.



*Change the column width by selecting columns, choosing **Format | Column Width**, and then entering a number into the **Column Width** dialog. If you are having difficulty with hidden columns, select all columns surrounding the hidden column and use **Edit | Column Width**.*

Alternatively, you can change the column width with the mouse. When the cursor is moved to the line that defines the right boundary of the column header, the cursor changes to a line with two arrows.  Press and hold the left mouse button and drag the cursor to the left or right to change the width of the column.

You can hide a column by moving the right edge of the column to the left edge (i.e., by making the column width equal to zero). To display hidden columns, press and hold the left mouse button at the edge of the hidden column and move the cursor to the right to widen the column.


Row Height

You can change the row height of selected cells by typing a value into the **Row Height** box, or by using the mouse to resize the row. The Excel Spreadsheet [.XLS] or Microsoft SYLK [.SLK] file format must be used to save the row height information within the file, since ASCII files such as [.CSV], [.TXT], [.DAT], [.BLN], and [.BNA] do not save format information.

To set row heights or to hide rows, select the entire row or individual cells within the rows, and then select **Format | Row Height**. For information on selecting rows, see the *Tips on Selecting Cells* section in this chapter. Enter the width for the selected rows or cells in the **Row Height** dialog. Rows can range from zero to 512 characters in height. A value of zero hides the row.



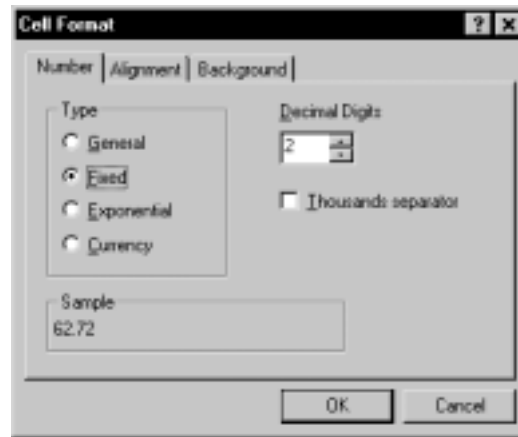
*Change the row height by selecting rows, choosing **Format | Row Height**, and then entering a number into the **Row Height** dialog. If you are having difficulty with hidden rows, select all rows surrounding the hidden row and use **Edit | Row Height**.*

Alternatively, you can change the row height with the mouse. When the cursor is moved to the line that defines the lower boundary of the row header, the cursor changes to a line with two arrows.  Press and hold the left mouse button, and drag the cursor up or down to change the height of the row. Hide a row by moving the cursor up until the next dividing line is reached. To display hidden rows, position the cursor at the bottom of the hidden row and drag the cursor down to increase the row height.

Number Format

Choose **Format | Cell Properties** to change the numeric data display in the worksheet. Number formatting has no effect on a numeric text string (numbers entered as text). For example, an ASCII data file might contain the digits "8123" (digits surrounded by quotes), numbers with letters, or numbers with symbols (i.e. a backslash "\"). These "numbers" are read as text and not as a number. See *Data as Numbers or Text* in this chapter for additional information.

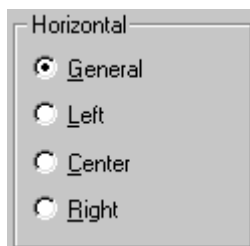
On the **Number** page, the *General* option displays numbers as precisely as possible within the available cell width. The *Fixed* option displays numbers as "d.ddd." The *Exponential* option displays numbers as "d.ddde+ddd." You can set the number of digits to the right of the decimal in the *Decimal Digits* box with the *Fixed* and *Exponential* options. The *Currency* option displays fixed numbers with a currency symbol such as the dollar sign (\$). If the *Thousands Separator* box is checked, a comma appears every three digits to the left of the decimal point.



These are the number format options located on the **Number** page accessed through **Format | Cell Properties**.

Alignment

Entries in a cell can be aligned in one of four ways. By default, imported ASCII files automatically align numbers to the right and text to the left. On the **Align** page, the *General* option aligns text to the left side of the cell, and numbers on the right side of the cell. A number with an apostrophe in front of it ('8123) is a text string. The apostrophe only shows in the active cell edit box. See *The Worksheet Window* section of this chapter for the location of the active cell edit box. The *Left* option aligns text and numbers to the left side of the cell. The *Center* option centers text and numbers within the cell. The *Right* option aligns text and numbers to the right side of the cell.



You can use the worksheet cell alignment options to change the display of your data.

Background Color

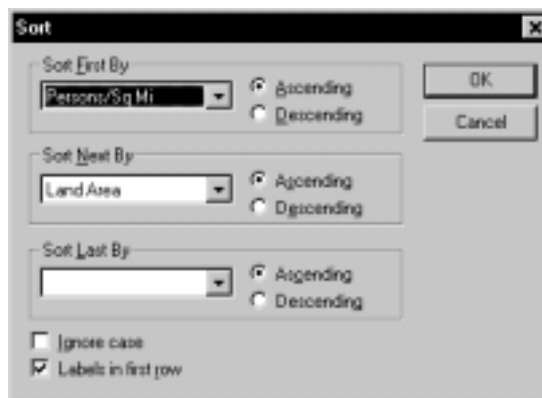
You can click on a color to set the cell background color on the **Background** page. Click *None* to remove any previously assigned background colors. The worksheet must be saved in the Excel Spreadsheet [.XLS] format to save the background color information in the file.

Formatting the Data

Selected cells of data are sorted or transformed using the **Data** menu commands. The **Data** menu also contains a statistics option. Various statistical results generated from the data can be saved in the worksheet.

Sort

The **Data | Sort** command arranges data according to rank in user-specified sort columns. Sorting rank is based on numbers, ASCII characters, and punctuation. You can sort numeric data, text, or mixed columns.



*You can use the **Data | Sort** command to sort data on multiple columns.*

Chapter 3 - Data Files and the Worksheet

Selecting Cells to Sort

Sorting is performed only on the selected columns (see *Tips on Selecting Cells*). If only one column is selected, only that column is sorted. To keep records (rows of data) together, select all columns containing data even if only one column is sorted.

Sort Order

The *Sort First By* option defines the primary column on which the rows are sorted. The positions of the sorted rows are determined by the *Ascending* or *Descending* rank in the *Sort First By* column.

Secondary Sort

When two or more rows have identical entries in the *Sort First By* column, the *Sort Next By* column can further organize the data set. Duplicates in the *Sort First By Column* are then sorted according to the rank in the *Sort Next By* column.

Final Sort

The *Sort Last By* column can be used when the *Sort Next By* column contains duplicates.

Ascending or Descending Sort

The sort order in an *Ascending* sort is based on the ASCII table. Numeric values are placed first, followed in order by cells starting with a space character, common punctuation, numeric text (numbers entered as text), uppercase letters, less common punctuation, lower case letters, uncommon punctuation, and blank cells. *Descending* order is the opposite of ascending order although blank cells are still listed last.

0	1	2	3	4	5	6	7	8	9
space	!	"	#	\$	%	&	'	()
*	+	,	-	.	/	"0"	"1"	"2"	"3"
"4"	"5"	"6"	"7"	"8"	"9"	:	;	<	=
>	?	@	A	B	C	D	E	F	G
H	I	J	K	L	M	N	O	P	Q
R	S	T	U	V	W	X	Y	Z	[
\]	^	_	`	a	b	c	d	e
f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y
z	{		}	~	blank				

This table shows the sort order in the worksheet.

Ignore Case

Because sorting is based on an ASCII table, upper and lowercase letters are treated differently. For example, "A" is sorted separately from "a." If the letters are to be treated as the same during the sort, check the *Ignore case* option. When this check box is activated, "A" is considered identical to "a" in the sorting rank.

Labels in First Row

The data set may contain text identifying the data in the column (header information) in Row 1. In this case, click the *Labels in First Row* option, to exclude the label row from the sort process.

Transform

Use the **Data | Transform** command to apply mathematical transformations to columns. Valid math operators include addition (+), subtraction (-), multiplication (*), and division (/) as well as a large library of built-in mathematical functions. See *Appendix A, Mathematical Functions* for a list of functions. Parentheses should be used to override precedence. Parentheses can also be used for clarification.

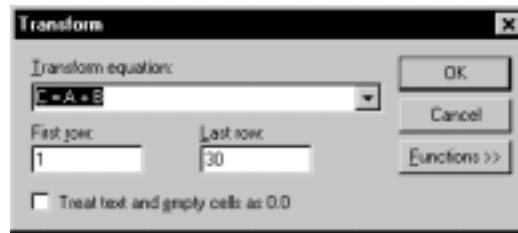
Type the formula you would like to use into the *Transformation equation* box. Formulas consist of a destination column on the left side of the equation and a mathematical manipulation on the right side of the equation. Use the column label letters on both sides of the equation. You can click the down arrow to use previously entered equations. By default, if any cell contains text or is empty, the formula is not calculated for that row. If you check *Treat text and empty cells as 0.0*, the text or empty cell is replaced with 0.0 and the numeric result is placed in the destination column. You can enter numbers into *First row* and *Last row* boxes to limit the calculation to the specified rows.

The *Functions* button opens a list of pre-defined mathematical functions. To use pre-defined functions, click the function in the list, click the *Insert* button, and then change the variable (X) in the function to a column letter in the transformation equation.

Chapter 3 - Data Files and the Worksheet

Example 1

Consider the simple example shown in the figure to the right. The *Transformation equation* is $C = A + B$. The *First row* is 1 and the *Last row* is 30. This transformation adds columns A and B and places the sum in column C for each of the 30 rows.

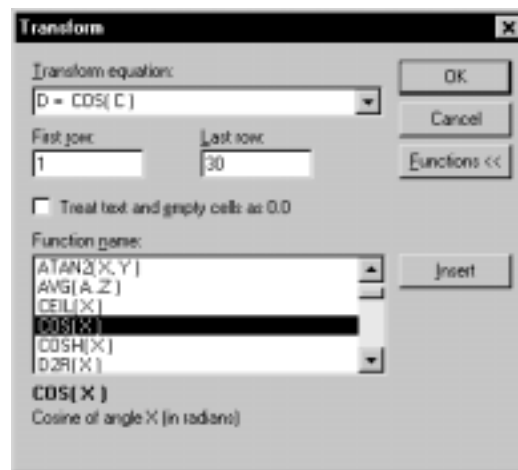


You can perform math in the worksheet by selecting the **Data / Transform** command.

Example 2

This example shows how to use the built in functions. Consider, for example, taking the cosine of data in Column C. Column D is the first empty column, so we will use Column D as the destination column.

1. Click on the **Data** menu and then select **Transform**. (You do not need to highlight any columns before selecting **Transform**.)
2. Type "D =" into the **Transform** equation box.
3. Click the *Functions* button.
4. Double-click on the function name COS(X) in the *Function name* group.
5. COS(X) is automatically placed in the equation. D = COS(X).
6. Replace the X in the function with the column letter containing the data to be transformed (Column C). D = COS(C).
7. Change the *First row* and *Last row* if you wish, and then press OK to create a new data column with Column C's data transformed with the cosine.



This is how the **Transform** dialog appears after following the steps to the left.

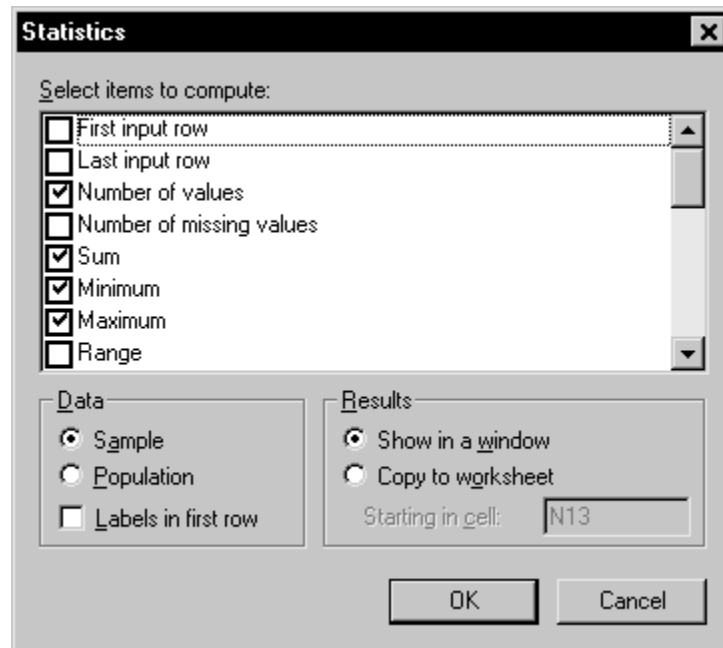
Statistics

The **Data | Statistics** command calculates statistical values for a group of selected numeric cells. For more information on selecting cells, see the *Tips on Selecting Cells* section of this chapter. Before using the **Statistics** command, select an entire column or a continuous group of cells in columns. If a rectangular block of rows and columns are selected, the **Statistics** command calculates the statistics for each column separately. A warning message appears if a group of cells cannot be used with the **Statistics** command. Non-numeric cell entries (empty cells or text) are excluded in statistics calculations.

The Statistics Dialog

Click in the check boxes next to the statistics options to calculate the statistics for the selected data.

- Select *Sample* or *Population* statistics in the *Data* group, depending on whether the data represent a statistical sample or the complete set of all possible members of a population.
- Check the *Labels in First Row* box if the first row of the selection contains descriptive labels. If this box is checked, the label appears at the top of the statistics report for each column.
- Select *Show in Window* to write the statistics results to a window. You can copy the results in this window to the clipboard to paste to other locations.
- Select *Copy to Worksheet* to write the data to part of the worksheet. Use the *Starting in Cell* box to specify the cell for the upper left corner of the statistics report. If the specified destination cells contain data, a warning is displayed that data will be overwritten. Click OK to overwrite the data or click *Cancel* to set a new *Starting in Cell* location.



Click in the box adjacent to the statistics name to compute the statistics for the selected columns.

Chapter 3 - Data Files and the Worksheet

Statistics Options

Statistic	Description
<i>First input row</i>	The <i>First input row</i> reports the first row number in the selection. If the <i>Labels in First Row</i> option is checked, the <i>First Input Row</i> is the second row in the selection.
<i>Last input row</i>	The <i>Last input row</i> reports the last row number containing data in the column.
<i>Number of values</i>	The <i>Number of values</i> indicates the number of numeric cells in the column.
<i>Number of missing values</i>	The <i>Number of missing values</i> indicates the number of non-numeric cells in the selection. If columns are selected by pressing the column letters, the number of missing values includes blank values up to the last used row in the worksheet, which may be different from the last-used row in the selected column. If cells are selected by highlighting specific cells, then only the blank cells within the selection are counted.
<i>Sum</i>	The <i>Sum</i> is the sum of all numeric cells in the column.
<i>Minimum</i>	The <i>Minimum</i> indicates the minimum value in the column.
<i>Maximum</i>	The <i>Maximum</i> indicates the maximum value in the column.
<i>Range</i>	The <i>Range</i> indicates the range of the numeric values in the column (<i>Maximum</i> – <i>Minimum</i>).
<i>Mean</i>	<p>The <i>Mean</i> is the arithmetic average of the data values. The mean is the sum of the data values divided by the number of data values.</p> $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ <p>where: n = number of data values x_i = i^{th} data value</p>
<i>Median</i>	The <i>Median</i> is the middle value among the data values. Half of the data values are larger than the median and half are smaller than the median. When there is an even number of data values, the median is the average of the two middle values.

Statistic	Description
<i>First quartile (25th percentile)</i>	The <i>First quartile (25th percentile)</i> is the value such that one-fourth of the data values are smaller than the quartile and three-fourths of the data values are larger than the first quartile.
<i>Third quartile (75th percentile)</i>	The <i>Third quartile (75th percentile)</i> is the value such that three-fourths of the data values are smaller than the quartile and one-fourth of the data values are larger than the third quartile.
<i>Standard error of the mean</i>	<p>The <i>Standard error of the mean</i> is an estimate of the standard deviation of means that would be found if many samples of n items were repeatedly collected from the same population.</p> <p>Suppose many samples of size n were repeatedly collected from the same population and the means of these many samples were calculated. The means of the samples would themselves form a data set. The standard error of the mean is an estimate of the standard deviation of this theoretical sample of means.</p> $SE = s / \sqrt{n}$ <p>where: s = sample standard deviation n = number of data values (for a sample)</p>

Chapter 3 - Data Files and the Worksheet

Statistic	Description
95% and 99% confidence interval for the mean	<p>If CI is the value of the confidence interval reported by the worksheet, the range of values between the sample mean minus CI and the sample mean plus CI is expected to include the true mean of the underlying population 95% of the time (for the 95% confidence interval) or 99% of the time (for the 99% confidence interval). This formula assumes that the data set is sufficiently large for the central limit theorem to apply.</p> <p><u>95% Confidence Interval for the Mean</u></p> $\pm t_{(n-1), \alpha=.05} (SE)$ <p><u>99% Confidence Interval for the Mean</u></p> $\pm t_{(n-1), \alpha=.01} (SE)$ <p>where: $t_{v, \alpha}$ = the value of the Student's t distribution with v degrees of freedom such that difference between the cumulative probability function evaluated at $t(v, \alpha)$ and $-t(v, \alpha)$ is equal to $1 - \alpha$ SE = standard error of the mean</p>

Statistic	Description
Variance	<p>The population <i>Variance</i> is the average of the squared deviations from the mean. The sample variance is the sum of the squared deviations from the mean divided by one less than the number of data values.</p> <p><u>Population Variance</u></p> $\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$ <p><u>Sample Variance</u></p> $s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$ <p>where:</p> <p>μ = population mean</p> <p>\bar{x} = sample mean</p> <p>N = number of data values (for a population)</p> <p>n = number of data values (for a sample)</p> <p>x_i = i^{th} data value</p>

Statistic	Description
<i>Average deviation</i>	<p>The <i>Average deviation</i> is the average of the difference between the absolute values of data points and the mean.</p> <p><u>Population Mean Deviation (MD)</u></p> $MD = \frac{1}{N} \sum x_i - \mu $ <p><u>Sample Mean Deviation (MD)</u></p> $MD = \frac{1}{N} \sum x_i - \bar{x} $ <p>where:</p> <p>μ = population mean</p> <p>\bar{x} = sample mean</p> <p>N = number of data values (for a population)</p> <p>n = number of data values (for a sample)</p> <p>x_i = i^{th} data value</p>
<i>Standard deviation</i>	<p>The <i>Standard deviation</i> is the square root of the variance.</p> <p><u>Population Standard Deviation</u></p> $\sigma = \sqrt{\sigma^2}$ <p><u>Sample Standard Deviation</u></p> $s = \sqrt{s^2}$ <p>where:</p> <p>σ^2 = population variance</p> <p>s^2 = sample variance</p>

Statistic	Description
<i>Coefficient of variation</i>	<p>The <i>Coefficient of variation</i> is the standard deviation divided by the mean. The worksheet reports the quotient; it does not convert the value to a percentage. The coefficient of variation is a dimensionless measure of variation. This statistic is not defined for the case of a zero mean and this measure is only useful when dealing with strictly positive data.</p> <p><u>Population Coefficient of Variation</u></p> $V = \sigma / \mu$ <p><u>Sample Coefficient of Variation</u></p> $V = s / \bar{x}$ <p>where:</p> <p>σ = population standard deviation</p> <p>s = sample standard deviation</p> <p>μ = population mean</p> <p>\bar{x} = sample mean</p>

Statistic	Description
<i>Coefficient of skewness</i>	<p><i>Coefficient of skewness</i> is a measure of asymmetry in the distribution. A positive skew indicates a longer tail to the right, while a negative skew indicates a longer tail to the left. A perfectly symmetric distribution, like the normal distribution, has a skew equal to zero. For small data sets, this statistic is an unreliable estimate of the population coefficient of skewness.</p> <p><u>Population Skew</u></p> $\gamma_1 = \frac{1}{N\sigma^3} \sum_{i=1}^N (x_i - \mu)^3$ <p><u>Sample Skew</u></p> $g_1 = \frac{n}{(n-1)(n-2)s^3} \sum_{i=1}^n (x_i - \bar{x})^3$ <p>where:</p> <p>σ = population standard deviation s = sample standard deviation μ = population mean \bar{x} = sample mean N = number of data values (for a population) n = number of data values (for a sample) x_i = i^{th} data value</p>

Statistic	Description
<i>Coefficient of kurtosis</i>	<p>The <i>Coefficient of kurtosis</i> is a measure of the data peak sharpness. Traditionally the value of this coefficient is compared to a value of 0.0, which is the coefficient of kurtosis for a normal distribution (i.e. the bell-shaped curve). A value greater than zero indicates a peaked distribution and a value less than zero indicates a flat distribution. Without a very large sample size, the use of this coefficient is of questionable value.</p> <p><u>Population Kurtosis</u></p> $\gamma_2 = \left(\frac{1}{N\sigma^4} \sum_{i=1}^N (x_i - \mu)^4 \right) - 3$ <p><u>Sample Kurtosis</u></p> $g_2 = \left(\frac{n(n+1)}{(n-1)(n-2)(n-3)s^4} \sum_{i=1}^n (x_i - \bar{x})^4 \right) - \frac{3(n-1)^2}{(n-2)(n-3)}$ <p>where:</p> <p>σ = population standard deviation</p> <p>s = sample standard deviation</p> <p>μ = population mean</p> <p>\bar{x} = sample mean</p> <p>N = number of data values (for a population)</p> <p>n = number of data values (for a sample)</p> <p>x_i = i^{th} data value</p>
<i>Kolmogorov-Smirnov goodness of fit for normal distribution</i>	<p>The <i>Kolmogorov-Smirnov goodness of fit for normal distribution</i> measures how well the data match a normal probability distribution. The K-S statistic is the largest difference between a normal distribution (with a mean equal to the sample mean, \bar{x}, and a standard deviation equal to the sample standard deviation, s) and the actual frequency distribution of the data.</p>

Chapter 3 - Data Files and the Worksheet

Statistic	Description
<i>Critical value of K-S statistic at 90%, 95%, or 99% significance level</i>	The <i>Critical value of K-S statistic at 90% significance level, Critical value of K-S statistic at 95% significance level, Critical value of K-S statistic at 99% significance level</i> are indicators of normal distributions. For example, if a sample collected from a population has a normal frequency distribution, the K-S statistic for that sample is less than the critical value 90, 95, or 99 percent of the time. If the K-S statistic is larger than the critical value, the hypothesis that the underlying population is distributed normally with a mean of \bar{x} and a standard deviation of s should be rejected.

Statistics References

Gilbert, Richard O., (1987), *Statistical Methods for Environmental Pollution Monitoring*, Van Nostrand Reinhold, New York.

King, Ronald S. and Bryant Julstrom, (1982), *Applied Statistics Using the Computer*, Alfred Publishing Company, Sherman Oaks, California.

Press, William H. et al, (1992), *Numerical Recipes in C*, Cambridge University Press, New York.

Spiegel, Murray R. (1997), *Schaum's Outline of Theory and Problems of Statistics*, McGraw-Hill, New York.

Sokal, Robert R. and F. James Rohlf, (1981), *Biometry: Principles and Practices of Statistics in Biological Research*, Freeman and Co, New York.

Zar, Jerrold H., (1974), *Biostatistical Analysis*, Prentice Hall, Englewood Cliffs, New Jersey.

Saving Data Files

If the file has already been saved, **File | Save** updates the saved file. If the file has not been saved previously, the **Save As** dialog appears when **Save** is selected from the **File** menu. If you would like to save an existing file to a new file name, choose **File | Save As** instead of **File | Save**.

In the **Save** or **Save As** dialog, you can type a new name into the *File name* box to save a new file. Also, the path and file name can be entered into the *File name* box instead of navigating through folders in the *Save in* field. Select the file format for the file in the *Save as type* list. You can save in several formats in the worksheet. Lotus files can be opened into the worksheet, but they cannot be saved in the Lotus formats.

Use Caution when Saving Excel Files!

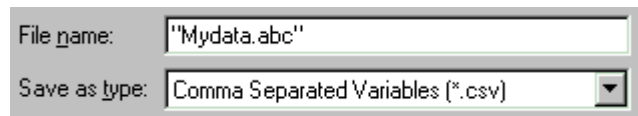
Surfer does not allow for saving multiple worksheets in a single Excel workbook. If a multi-worksheet Excel workbook is opened and saved as an [.XLS] file in **Surfer**, be aware that only the single worksheet is saved in the document. All the unused worksheets are lost. In this case, a warning message is issued. To avoid overwriting the file containing multiple sheets click *No* in the warning message box and save the file to a different file name.

File Names, Formats, and File Extensions

When a worksheet file is saved, the file format is specified by entering the appropriate extension on the file name. If the needed file is an ASCII DAT file, enter a file name such as MYDATA.DAT. The [.DAT] extension tells the worksheet to save the file in the ASCII DAT file format.

If the extension is not included in the file name, the format is determined by the dialog's *Save as type* field. For example, if the name MYDATA is entered into the *File name* field and the *Save as type* field is set to *Excel Spreadsheet [.XLS]*, the file is saved as MYDATA.XLS in the Excel format.

The file can be saved with any extension by enclosing the file name in double quotes. The file is saved with the name and extension entered into the *File name* field, but saved in the format



File name: "Mydata.abc"

Save as type: Comma Separated Variables (*.csv)

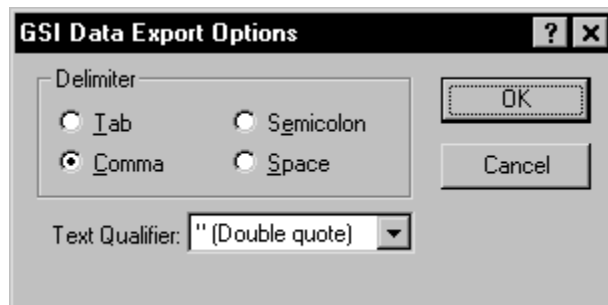
Enter a file name with quotes to preserve a file extension not available in the save as type field. The file above will be a [.CSV] file with an [.ABC] extension.

specified in the *Save as type* field. For example, enter the name (with quotes) "MYDATA.ABC" into the *File name* field. If the *Save as type* field is set to *Comma Separated Variables [.CSV]*, the file is saved as MYDATA.ABC in the [.CSV] format.

Saving ASCII Data Files

When a file is saved in the ASCII Golden Software Data [DAT] format, the **GSI Data Export Options** dialog is displayed with the following options:

- *Delimiters* are the characters used between cells in a single row (fields in a record), and can be commas, spaces, semicolons, or tabs.
- *Text Qualifiers* are double-quotes, single-quotes, or none. For example, if double-quotes are chosen, all non-numeric or mixed alphanumeric cell entries are surrounded by double-quotes in the file.



*Choose one Delimiter and a Text Qualifier for Golden Software Data files [DAT] in the **GSI Data Export Options** dialog.*

[.TXT] files are tab delimited ASCII text files with no quotes around the text strings. [.CSV] (comma separated variables) files are comma delimited with double-quotes around text strings (non-numeric or mixed alphanumeric).

Using Commas or Semicolons in Addition to Quotes

Although double quotes are not required around text strings, they are useful when creating a space-delimited file that contains text. Often there are text strings that contain spaces, as in a date containing month, day, and year. With space-delimited files, this single entry is interpreted as three cells when loading this file into the worksheet. The safest way to eliminate this problem is to place double quotes around all text strings and use comma delimiting between variables.

ASCII File Formatting

ASCII data files do not retain cell formatting such as background color. Use the Excel Spreadsheet [.XLS] format if you wish to retain cell formatting.

Page Setup

Before printing the worksheet, the page format of the worksheet can be set through **File | Page Setup**. **Page Setup** controls items such as printing headers and footers, centering the data on the page, showing gridlines, etc. There are three tabs in the **Page Setup** dialog, **Page**, **Margins**, and **Options**. There is a *Printer* button at the bottom of the dialog that allows you to set the default printer no matter what tab is open.

Page

You can set the paper *Size* and *Source* for the active printer in the *Paper* group. The active printer can be changed by pressing the *Printer* button at the bottom of the **Page Setup** dialog. The orientation of the printed page can be changed to *Landscape* or *Portrait* mode in the *Orientation* group.

The options in the *Scaling* group box control the printed size of the worksheet. The *Adjust to ___ % full size* option specifies the percent of full size that the worksheet prints. The arrow buttons can be used to scroll up or down from 100% (full size), or values can be typed into the box. The *Adjust to ___ % full size* setting is independent of the *Fit to* option.

The amount of data in the worksheet determines how many pages are required to print the worksheet. The *Fit to ___ page(s) across by ___ page(s) down* option tells the program to print the worksheet at 100% scale or less. This option does not automatically scale the printed worksheet greater than 100%. This option is most useful when the worksheet is large and the number of printed pages needs to be limited.



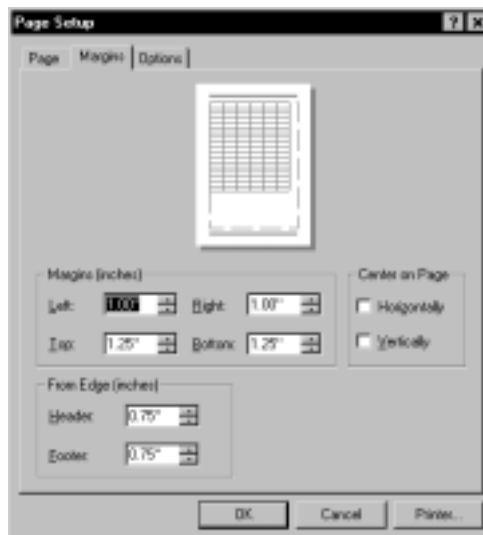
Page shows the page layout at the top (margins, orientation), includes options for paper sizes, orientation, and worksheet print scaling.

Margins

You can use the **Margins** tab to set the page margins for the printed page. Set the *Left*, *Right*, *Top*, and *Bottom* values in inches or millimeters to any limits the printer allows. The *Margin* settings are independent of the settings used for *Headers* or *Footers*.

Headers and *Footers* are the only items that print outside of the margins. The *From Edge* options control how far the *Header* or *Footer* is printed from the edge of the page. If these values are greater than the *Top* or *Bottom* margins, the worksheet data can print over the header or footer. The text printed for the header and footer is set on the **Options** page.

The *Center on Page* group options centers the printout *Horizontally* and *Vertically*. If neither option is selected the worksheet prints in the upper left corner of the page.



You can set the margin distance, header and footer distance, and page centering on the **Margins** page.

Options

The *Print* group controls how the worksheet information is printed. Check the *Gridlines* check box to draw gridlines separating each column and row on the printout. Check the *Row and column headers* option to print the column letters and row numbers of the worksheet on the printout. If the cells contain color backgrounds (set from the **Format | Cell Properties** command), use the *Black and white* option to print the worksheet in black-and-white. If the *Black and white* option is not checked, colors print as shades of gray on a black and white printer.

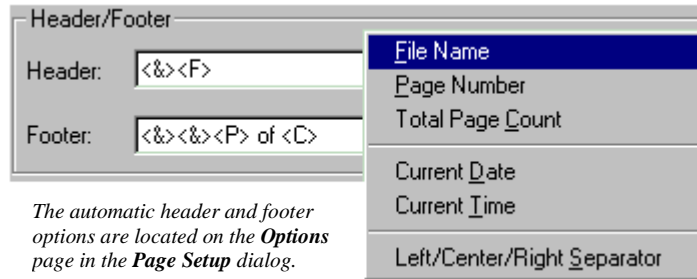
The *Page Order* group options control the order in which multiple pages are printed. The *Down and then across* option prints the worksheet from top to bottom first, and then moves to the right and prints top to bottom again. The *Across and then down* option prints from left to right first, and then moves down and prints left to right again.

The *Header/Footer* group options control the type of information included in the worksheet headers and footers in the printout. The *Header* appears at the top of the page, and the *Footer* appears at the bottom of the page. The headers and footers are spaced from the edge of the page based on the *From Edge* option of the **Margins** page. Descriptive text can be typed into the *Header* and *Footer* boxes, or click the arrows to the right of the boxes and select automatic options from the list.

Automatic Header/Footer Codes

- *File Name* <F> prints the name of the active file. The drive and path are not included.
- *Page Number* <P> prints the page number on each page. When several pages are printed, the order of printing is controlled from the *Page Order* group.
- *Total Page Count* <C> prints the total number of pages required to print out the worksheet.
- *Current Date* <D> prints the current date.
- *Current Time* <T> prints the current time.
- *Left/Center/Right Separator* <&> separates the header and footer text. Too many separators can push text off the page. If this happens, remove the <&> separator and use spaces instead.

Chapter 3 - Data Files and the Worksheet



Examples

For a six page document, <&><&>Page <P> of <C> would print the page number of total pages right justified:

Page 1 of 6

Enter Joe Smith<&><F><&><D> to print out a name, file name, and date:

Joe Smith

COLORADO.DAT

9/11/97

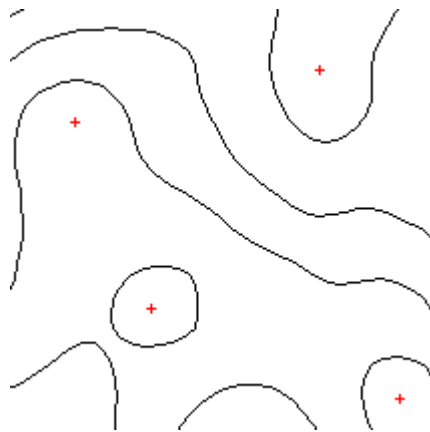
Creating a Data File with Digitize

In the plot window, the **Map | Digitize** command can be used to create a data file from map XY coordinates. The **Digitize** command is used to collect coordinates from the map, and create a [.BNA] or [.BLN] file. See *Appendix C, File Formats* for more information on the [.BNA] and [.BLN] file formats.

Select a map to activate the **Digitize** command. If anything other than a map is selected, (i.e. text boxes, composite objects, drawing objects) the **Digitize** command is not active.

To digitize points from a map:

1. Select a map.
2. Select **Map | Digitize**.
3. The arrow pointer changes into a cross hair pointer.
4. Digitize points by clicking with the left mouse button on the map. You can zoom in on the map for greater precision in digitizing.
5. The digitized points appear as temporary small red crosses in the window. The digitized point coordinates appear as text in a report window.
6. Save the contents of the report window to save the digitized points.



Digitized points appear as temporary small red crosses on the map.

Worksheet Technical Specifications

Worksheet Size and Memory Requirements

In a worksheet, the maximum number of rows is one billion, and the maximum number of columns is one billion. The approximate memory requirements for unformatted numeric data are 10.5 bytes per cell.

Numeric Limits

Numeric data are stored in the worksheet as double precision floating-point numbers. The largest number that can be represented is approximately 1.79769E+308. The smallest number that can be represented is approximately -1.79769E+308.

The smallest positive number that can be represented is 2.22507E-308. However, double precision floating-point numbers can only represent approximately 15 significant decimal digits.

Chapter 4

Creating Grid Files

Introduction to Grid Files

Contour maps, image maps, shaded relief maps, vector maps, wireframes, and surfaces all require grids for their generation in **Surfer**. A grid is a regular, rectangular array of values. The **Grid | Data** command provides you with several methods for generating a grid file [.GRD] from your XYZ data. **Surfer** can also use USGS [.DEM], GTopo30 [.HDR], SDTS [.DDF] or DTED [.DT*] grid files directly. This means that you do not have to go through the gridding process if you already have a [.DEM], [.HDR], [.DDF], or [.DT*] file.

Gridding methods produce a regularly spaced, rectangular array of Z values from irregularly spaced XYZ data. The term "irregularly spaced" means that the points follow no particular pattern over the extent of the map, so there are many "holes" where data are missing. Gridding fills in these holes by extrapolating or interpolating Z values at those locations where no data exists.

A grid is a rectangular region comprised of evenly spaced rows and columns. The intersection of a row and column is called a grid node. Rows contain grid nodes with the same Y coordinate, and columns contain grid nodes with the same X coordinate. Gridding generates a Z value at each grid node by interpolating or extrapolating the data values.

Grid file columns and rows are sometimes referred to as X grid lines and Y grid lines, respectively. In fact, when you produce a wireframe from a grid file, the lines that make up the map actually represent the X and Y grid lines. The intersection of the grid lines define the location of the grid nodes on the wireframe, and the height of the surface at each grid node is proportional to the Z value assigned to that node.

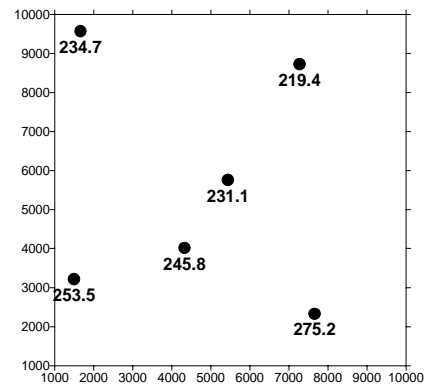
When XYZ data are collected on regular intervals, you may produce a grid file that uses the Z values directly and does not interpolate the values for the grid nodes. These procedures are explained in the *Producing a Grid File from a Regular Array of XYZ Data* section of this chapter.

A Gridding Example

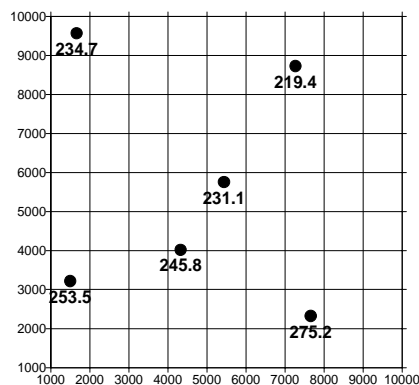
Consider the problem of producing a contour map of water table depth given well data collected over a region. The well locations are not regularly spaced over the area of interest. If you provide **Surfer** with the locations of the wells (the XY coordinates) and the depth to the water table (the Z value) in the form of an XYZ data file, **Surfer** can produce a grid file and, ultimately, a grid-based map from the original data. The following series of figures show the normal progression from a data file, to a grid file, to a contour map.

	A	B	C
1	X Coord.	Y Coord.	Z value
2	1665.4	9567.2	234.7
3	7659.3	2324.6	275.2
4	1499.5	3212.9	253.5
5	5438.1	5753.9	231.1
6	4327.4	4013.9	245.8
7	7272.4	8721.9	219.4

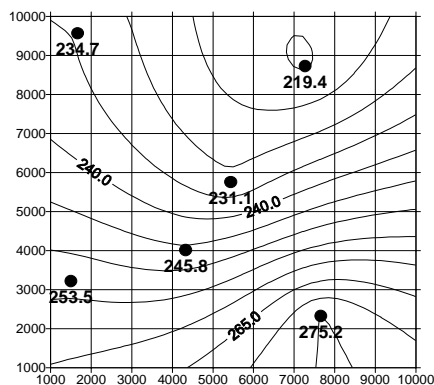
1. This is the XYZ data file that defines the well locations and water table depth at each location.



2. This is a post map displaying the data locations. The well locations are irregularly spaced over the map in this example. There are many "holes" where no data exists.



3. Gridding interpolates a Z value at the intersection of each row and column in the grid file, thereby filling holes in the data. Here the rows and columns are represented by grid lines drawn across the map.



4. The irregularly spaced data points are used to interpolate grid node values. These interpolated values are written to a grid file. The grid file is used to produce the contour map. This figure shows the contour map, the posted data points, and the layout of the grid.

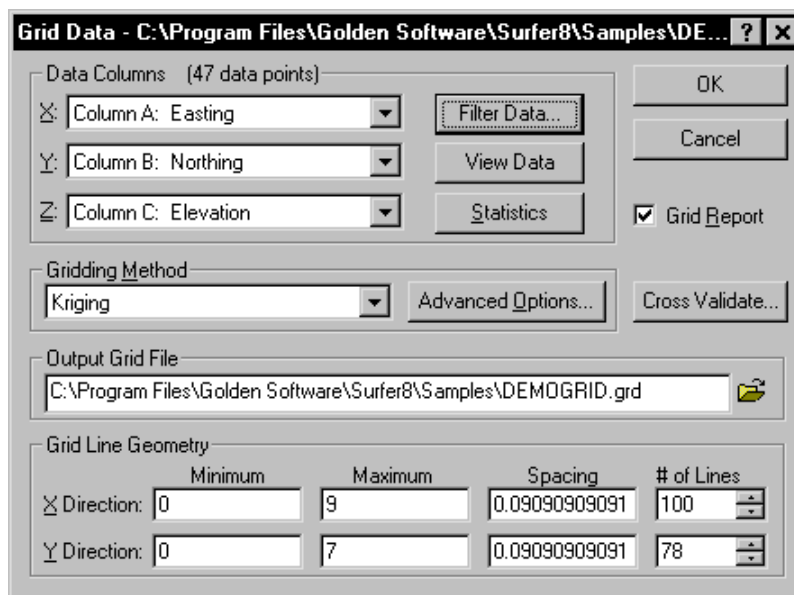
Creating a Grid File

Grid files are produced using the **Grid | Data** command. The **Data** command creates a grid file from an XYZ data file. When creating a grid file you can usually accept all of the default gridding parameters and generate a grid file that represents your data well. Under most circumstances, the recommended gridding method is kriging with the default linear variogram. This is the selected default gridding method because it gives good results for most XYZ data sets.

There are several gridding parameters you can set when producing a grid file. Each parameter is discussed in detail in this chapter.

To produce a grid file from an XYZ data file:

1. Create an XYZ data file. The data must be organized in columns: all X data in one column, all Y data in another column, and all Z data in a third column.
2. Choose the **Grid | Data** command to display the **Open** dialog.
3. Select the XYZ data file and click *Open*.
4. In the **Grid Data** dialog, specify the parameters for the type of grid file you want to produce.
5. Click OK and the grid file is created. During gridding, the status bar at the bottom of the **Surfer** window provides you with information about the gridding process.



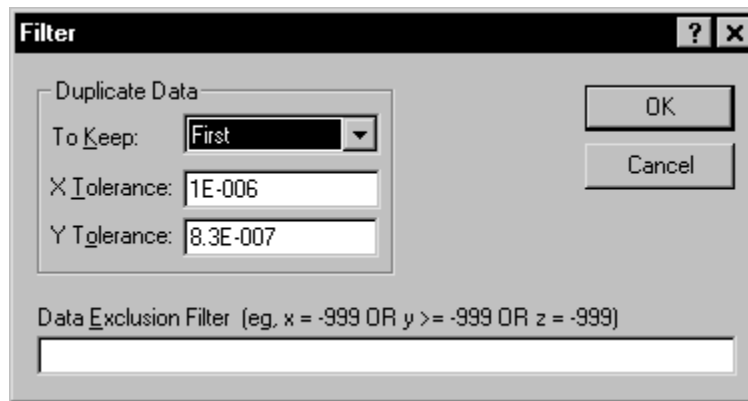
Gridding options are set through the **Grid Data** dialog.

Data

All gridding methods require at least three non-collinear data points, organized into XYZ data files. Some methods require more data points. For example, a higher-order polynomial fit needs more than three data points because there must be at least as many data points as there are degrees of freedom. Select **Grid | Data** to choose the data to be used in the gridding process.

Filter Data

You can filter the data before gridding based on a pre-defined filter or based on a user-defined equation. The **Filter** dialog provides methods for filtering data and it is opened by clicking the *Filter Data* button in the **Grid Data** dialog. Filtering is also available on the **Data** page in the **New Variogram** dialog. You can filter data by using one of the predefined data filtering methods (*To Keep* list), or you can generate a filtering method with the *Data Exclusion Filter*. When filtering data, the *Data Exclusion Filter* is applied first, and then the *Duplicate Data* methods are applied.



*You can open the **Filter** dialog by clicking the Filter Data button in the **Grid Data** dialog.*

Duplicate Data

The *Duplicate Data* group contains methods for defining and handling duplicate data points. Duplicate data are two or more data points having nearly identical XY coordinates (Z values may vary for these XY coordinates). Duplicate data are a problem for a number of gridding methods, including kriging, natural neighbor, radial basis function, and triangulation with linear interpolation.

To Keep

Duplicates are determined by moving from the lowest X value to the highest X value. A datum only belongs to one set of duplicates. The *To Keep* options specify which duplicate data points to keep and which to delete in each set of duplicate points. A report containing the retained and deleted points can be generated.

To Keep	Description
<i>All</i>	Do not delete any duplicates. This option is not available for kriging, nearest neighbor, natural neighbor, radial basis function, and triangulation with linear interpolation.
<i>None</i>	Eliminate all of the duplicates.
<i>First</i>	Keep the first point, as defined by the order in the data file, from each set of duplicates.
<i>Last</i>	Keep the last point, as defined by the order in the data file, from each set of duplicates.
<i>Minimum X</i>	Keep the point with the minimum X coordinate.
<i>Maximum X</i>	Keep the point with the maximum X coordinate.
<i>Median X</i>	Keep the point with the median X coordinate.
<i>Minimum Y</i>	Keep the point with the minimum Y coordinate.
<i>Maximum Y</i>	Keep the point with the maximum Y coordinate.
<i>Median Y</i>	Keep the point with the median Y coordinate.
<i>Minimum Z</i>	Keep the point with the minimum Z value.
<i>Maximum Z</i>	Keep the point with the maximum Z value.
<i>Median Z</i>	Keep the point with the median Z value.
<i>Sum</i>	Create an artificial datum at the centroid of the duplicate points with a Z value equal to the sum of the duplicate set's Z values.
<i>Average</i>	Create an artificial datum at the centroid of the duplicate points with a Z value equal to the average of the duplicate set's Z values.
<i>Midrange</i>	Create an artificial datum at the centroid of the duplicate points with a Z value equal to the midrange of the duplicate observations' Z values halfway between the minimum Z and the maximum Z.
<i>Random</i>	Keep a single, randomly selected, representative point.

Chapter 4 - Creating Grid Files

X and Y Tolerance

In addition to the *To Keep* options there are *X Tolerance* and *Y Tolerance* options. For example, two points, A and B are duplicates if:

$$|X_A - X_B| < X \text{ Tolerance}$$

and

$$|Y_A - Y_B| < Y \text{ Tolerance}$$

Using this definition, it is possible for points A and B to be duplicates, for point B and C to be duplicates, but for A and C to not be duplicates.

Excluding Data

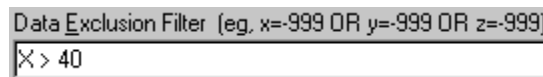
In addition to eliminating duplicates, you can also filter data based on Boolean functions. The *Data Exclusion Filter* allows you to use a Boolean expression to specify how to exclude data. For example:

Exclusion Filter	Purpose
$X = -999$ or $Y = -999$ or $Z = -999$	excludes any triplet with a -999 field
$X < 10$ or $X > 20$ or $Y < 10$ or $Y > 20$	excludes all data outside the range 10 to 20
$Z < 0.0$	excludes any triplet with $Z < 0.0$

Boolean expressions, used by **Grid | Function**, **Grid | Math**, **Grid | Data**, and **Grid | Variogram**, include:

- logical operators (and, or, xor, not)
- comparison operators ($=$, $<>$, $<$, $>$, $<=$, $>=$)
- IF function - e.g., IF(condition,true_value,false_value).

The words AND, OR, XOR, NOT, and IF are reserved keywords and may not be used as variables names.



*The Data Exclusion Filter is found in the **Filter** dialog and on the **Data** page in the New **Variogram** dialog.*

For example, consider the case of ignoring data outside of a grid. The original grid *X Maximum* is 50, but the grid *X Maximum* is reset to 40. To limit the search to data with X values less than 40, use the *Data Exclusion Filter* by entering $X > 40$ into the *Data Exclusion Filter* text box. This tells **Surfer** to exclude all data with X values greater than 40.

Statistics

Click the *Statistics* button to generate a statistics report based on the selected XYZ columns. Refer to the *Reports* topic in online help for detailed information on each of the reported statistics.

View Data

If you are unsure of which columns contain your XYZ data, click the *View Data* button to see the data file in a worksheet format. If you get an *Insufficient data (3 or more XYZ triplets required)* error, use *View Data* to make sure that all three columns of data are right aligned. If one of the columns is left aligned, the data are text, not numbers. Refer to *Data as Numbers or Text* in *Chapter 3, Data Files and the Worksheet* for more information.

Grid Line Geometry

The grid line geometry is set in the **Grid Data** dialog. Grid line geometry defines the grid limits and grid density. Grid limits are the minimum and maximum X and Y coordinates for the grid. Grid density is usually defined by the number of columns and rows in the grid. The *# of Lines* in the *X Direction* is the number of grid columns, and the *# of Lines* in the *Y Direction* is the number of grid rows. By defining the grid limits and the number of rows and columns, the *Spacing* values are automatically determined as the distance in data units between adjacent rows and adjacent columns.

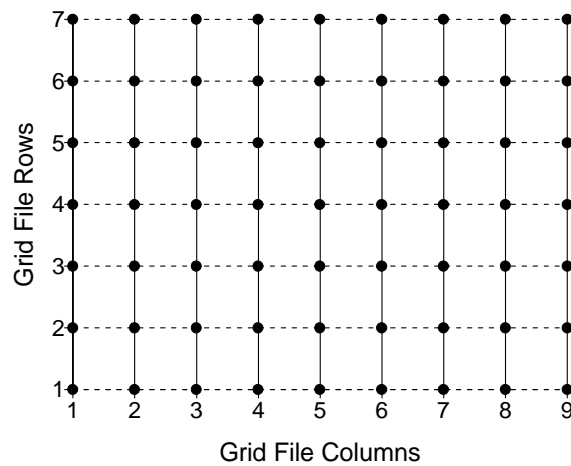
	Minimum	Maximum	Spacing	# of Lines
X Direction:	0	9	0.0909090909	100
Y Direction:	0	7	0.0909090909	78

*In the **Grid Data** dialog, the Grid Line Geometry group allows you to specify the grid limits and the grid density by entering the appropriate values.*

Surfer computes the minimum and maximum X and Y values from the XYZ data file. These values are used as the default minimum and maximum coordinates for the grid. The direction that covers the greater extent (the greater number of data units) is assigned 100 grid lines by default. The number of grid lines in the other direction is computed so that the grid line spacing in the two directions are as close to one another as possible.

Chapter 4 - Creating Grid Files

The simple example shown here demonstrates the relationship between grid rows, columns, and grid nodes. In this example, the grid spacing is 1.0 between rows and columns. The number of grid lines in the X direction (the number of columns in the grid file) is nine, and the number of grid lines in the Y direction (the number of rows in the grid file) is seven.



This is an example of a nine by seven grid. The grid columns are represented by vertical solid lines, and the grid rows are represented by horizontal dashed lines. The grid nodes are shown as dots at the intersection of the rows and columns.

Three examples are given in the table below. These examples show the default parameters generated based on the minimum and maximum X and Y values from the three XYZ data files.

Row/Column	Minimum	Maximum	Grid Line Spacing	# of Grid Lines
X Direction	0	99	1	100
Y Direction	0	79	1	80
X Direction	100	109	0.09090909	100
Y Direction	50	57	0.09090909	78
X Direction	-105.75	-105.38	0.00434343	86
Y Direction	39.6	40.03	0.00434343	100

Grid Limits

Grid limits define the extent of contour maps, image maps, shaded relief maps, vector maps, wireframes, and surfaces created from grid files. When creating a grid file, set the grid limits at the extents you want to use for your map. Once a grid file is created, you cannot produce a grid-based map larger than the extent of the grid file. If you find you need larger grid limits, you must regrid the data. You can, however, read in a subset of the grid file to produce a map smaller than the extent of the grid file.

To specify the grid limits:

1. Click **Grid | Data**.
2. Open a data file.
3. Examine the values in the *Grid Line Geometry* group. The *Minimum* and *Maximum* boxes for the *X Direction* and the *Y Direction* show the data limits for the file to be gridded.
4. You can set different limits by typing new values into the boxes. For example, if your data ranges from one to nine in the *X Direction*, the default limits in the *X Direction* are set at one for the *Minimum* and nine for the *Maximum*. If you want the grid file to range from zero to 10, you must set these limits by specifying the *Minimum* and *Maximum* values to use in the *X Direction*. The same applies for the *Y Direction*.
5. Specify any other gridding parameters, click OK, and the grid file is created with the limits you specify.

Grid Density

Grid density refers to the number of rows and columns in a grid file, and it is a measure of the number of grid nodes in the grid. Higher grid densities increase the smoothness in grid-based maps. However, an increase in the number of grid nodes proportionally increases the gridding time, drawing time, and the grid file size. You can have up to 10,000 rows and columns in a grid file.

Greater grid density generates visually smoother maps. Contour lines and lines defining a wireframe are a series of straight-line segments. More X and Y lines in a grid file result in shorter line segments for contours or wireframes. This provides a smoother appearance to contour lines on a contour map or smoother appearing wireframe.

Although highly dense grid files can be created, time and space are practical limits to the number of grid nodes you may want to create in a grid file. Grid density limits are based on the amount of available memory in your computer and the size of the data file used to create the grid. Limited memory, very large data files, very dense grids, or any combination of these factors can greatly increase gridding time. When gridding begins, the status bar provides you with information about

Chapter 4 - Creating Grid Files

the estimated gridding time to complete the task. If gridding time is excessive, click in the plot window to cancel the gridding operation.

You can increase or decrease the grid density by using the **Grid | Spline Smooth** command, or the **Grid | Extract** command. See *Chapter 18, Grid Operations* for more information on these commands.

To specify the grid density:

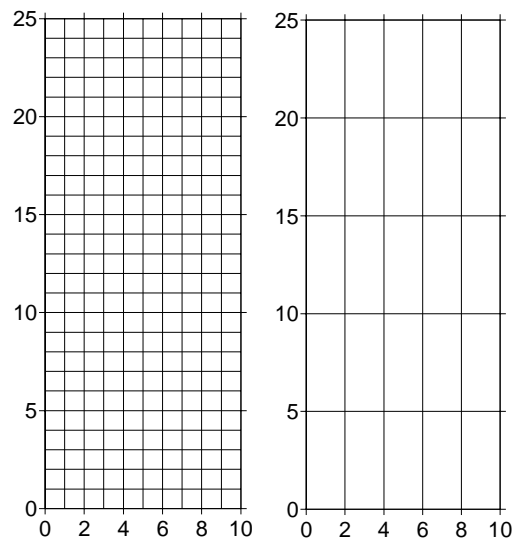
1. Click **Grid | Data**.
2. Open a data file.
3. You can specify *Spacing* values or *# of Line* values for both the *X Direction* and *Y Direction* in the *Grid Line Geometry* group in the **Grid Data** dialog. These values, in conjunction with the grid limits, control the grid density in the grid file.
 - The *Spacing* boxes specify the number of data units between grid lines. This value is a function of the *Minimum* and *Maximum* values and the *# of Lines* values. When you change these values, the *# of Lines* values are automatically recalculated to reflect the change. If the *Spacing* value you specify results in a fractional number for the *# of Lines* value, **Surfer** automatically adjusts the *Spacing* value slightly to generate an integer number for the *# of Lines* value.
 - The *# of Lines* boxes specify the number of lines in the X and Y directions. This is the number of rows (Y) and columns (X) generated in the grid, and must be a whole number. If you change the *# of Lines* values, the *Spacing* values automatically change to reflect the new values. If the data limits are different in the X and Y directions, the *Spacing* in *Grid Line Geometry* is unequal if the *# of Lines* values are equal.
4. Specify any other gridding parameters, click OK, and the grid file is created with the selected limits.

Examples

Consider these examples. The data range from 0 to 25 in the Y dimension and 0 to 10 in the X dimension. The two examples use different numbers of grid lines, or grid spacing, during gridding. The *Grid Line Geometry* information for both examples is given below.

In the example on the left, the grid *Spacing* is set approximately equal in the X and Y dimensions (one unit each). This results in a different number of grid lines in the X and Y dimensions.

In the example on the right, the same *# of Lines* are specified in the two dimensions. This results in an unequal spacing in data units in the two dimensions.



Two different Grid Line Geometry examples are shown here. These grids are based on the same data file. The coordinates range from zero to 10 in the X dimension and zero to 25 in the Y dimension.

Grid Line Geometry				
	Minimum	Maximum	Spacing	# of Lines
X Direction:	0	10	1	11
Y Direction:	0	25	1	26

This shows the Grid Line Geometry information for the 11 by 26 grid. The grid line spacing values are set to one, resulting in a different number of grid lines in the X and Y dimensions.

Grid Line Geometry				
	Minimum	Maximum	Spacing	# of Lines
X Direction:	0	10	2.5	5
Y Direction:	0	25	6.25	5

This shows the Grid Line Geometry information for the 5 by 5 grid. The number of lines is equal, resulting in different spacing in the X and Y dimensions.

Cross Validate

Generally, cross validation can be considered an objective method of assessing the quality of a gridding method, or to compare the relative quality of two or more candidate gridding methods. In **Surfer**, cross validation can be used with all gridding methods. While cross validation can be used to select a gridding method, the results can also be used to assess the spatial variation in gridding quality and to guide data sampling.

A generalized discussion of cross variation is given here. Refer to one of the many geostatistics books for more information. In the listed references, much of the discussion concerns kriging, but the generalized discussion applies to all of the gridding methods in **Surfer**.

The Cross Validation Process

Given the known values at N observation locations in the original data set, cross validation allows you to assess the relative quality of the grid by computing and investigating the gridding errors. In **Surfer**, these errors are calculated by removing the first observation from the data set, and using the remaining data and the specified algorithm to interpolate a value at the first observation location. Using the known observation value at this location, the interpolation error is computed as:

$$\text{error} = \text{interpolated value} - \text{observed value}$$

Then, the first observation is put back into the data set and the second observation is removed from the data set. Using the remaining data (including the first observation), and the specified algorithm, a value is interpolated at the second observation location. Using the known observation value at this location, the interpolation error is computed as before. The second observation is put back into the data set and the process is continued in this fashion for the third, fourth, fifth observations, etc., all the way through up to and including observation N. This process generates N interpolation errors. Various statistics computed for the errors can be used as a quantitative, objective measure of quality for the gridding method.

Thus, cross validation involves four steps:

1. Select a gridding method, along with all of the defining parameters.
2. For each observation location, interpolate the value using the neighboring data, but not the observation itself.
3. Compute the resulting interpolation errors.
4. Assess the quality of the selected gridding method using various summary statistics of the errors.

Using Cross Validation

To obtain cross validation information:

1. Select **Grid | Data**.
2. Select a data file in the **Open** dialog.
3. Set the *Gridding Method* and its *Advanced Options*.
4. Click the *Cross Validate* button in the **Grid Data** dialog.
5. Select the number of random points, the validation limits, and exclusion limits you wish to use with cross validation.
6. Click the *Report* box to generate a cross validation report.
7. Set the results path and file name in the *Cross Validation Results File* field.

The Cross Validation Dialog

Number of Random Points to Validate

For large and very large data sets the cross validation process of sweeping through each and every observation point can take a great deal of time and computational effort. A quicker alternative is to carry out the cross validation procedure at a random subset of the observation locations.

Cross Validation

Number of random points to validate: 47

Select validation points within these limits

	Minimum	Maximum
X Direction:	0	9
Y Direction:	0	7
Z Direction:	25	105

☒ Report

Exclude data within this distance of the validation point:

X Tolerance: 0 Y Tolerance: 0

Cross Validation Results File

C:\Program Files\Golden Software\Surfer8\Samples\Demogrid.V

OK Cancel

*Click the Cross Validate button in the **Grid Data** dialog to open the **Cross Validation** dialog.*

Chapter 4 - Creating Grid Files

Number of random points to validate allows you to specify the number of cross validation points. By default, this value is equal to the total number of observations. For large and very large data sets this values should be 1,000 or more to ensure a relatively stable set of error statistics. Note, the entire data set is used throughout the cross validation process, not just the random subset. The random subset merely identifies the locations at which cross validation errors are computed. Also, note that the random subset is determined without replacement, using a pseudo-random number generator that is randomly initialized for every execution; thus, the random subset may be different every cross validation run.

Select Validation Points Within These Limits

Set the *X Direction*, *Y Direction*, and *Z Direction* minimum and maximums in the *Select validation points within these limits* group to restrict the cross validation to a subarea of the data extent. Data falling outside of these limits may be used during the interpolation, but they are not used as cross validation points. The limits in the *Z Direction* are useful to exclude cross validation at known anomalous observation locations.

Exclude Data Within this Distance of the Validation Point

Often observations come in relatively homogeneous clusters. In these circumstances, the standard cross validation approach may not generate useful results, as the interpolated values are merely the values of the close-by adjacent observations. To counter this potential problem, the *X Tolerance* and *Y Tolerance* fields define a centered rectangular buffer zone around each of the observations at which cross validation is carried out. This buffer zone is two times the *X Tolerance* in the X direction, and two times the *Y Tolerance* in the Y direction. Any observations falling within a validation point's buffer zone are not used in the interpolation of that validation point.

Cross Validation Results File

Set a path and file name in the *Cross Validation Results File* field for the cross validation results. The results are presented in an ASCII data file, which can be used as a gridding data file. The first line in the file contains column titles. Each subsequent line is associated with a validation point. There are seven numeric values per line. The first three values are the X, Y, and Z values of the validation point. The fourth value, titled *ID* is the line number from the original data file of the validation point. The next two numbers are the estimated and residual values. The *nData* column contains the total number of original data points.

Statistics Note

Under most interesting statistical models, the cross validation errors are statistically dependent: the estimate at one observation location uses much of the same information as the estimate at a neighboring observation location. As such, standard hypothesis testing and test of statistical significance are not valid. A discussion of this point can be found in Kitanidis (1997).

Cross Validation References

Below are a few references for cross validation. The following references include extensive discussion of the theory and application of cross validation. Isaaks and Srivastava [1989], Kitanidis [1997], Olea [1999], and Chiles and Delfiner [1999] focus on cross validation with kriging, but the general discussion applies to all of the supported gridding methods in **Surfer**.

Chiles, J. P. and P. Delfiner. (1999) *Geostatistics: Modeling Spatial Uncertainty*. John Wiley and Sons, New York, 695 pp.

Isaaks, E. H. and R. M. Srivastava. (1989) *Applied Geostatistics*. Oxford University Press, Inc., New York, 561 pp.

Kitanidis, P. K. (1997) *Introduction to Geostatistics: Applications in Hydrogeology*. Cambridge University Press, Cambridge, 249 pp.

Olea, R. (1999) *Geostatistics for Engineers and Earth Scientists*. Kluwer Academic Publishers, Boston, 303 pp.

Grid Report

Check the *Grid Report* box in the **Grid Data** dialog to create a gridding report that includes all the gridding parameters used to generate a grid. This report also includes statistics. For detailed information about the contents of this report, see the online help topic *Reports*.

General Gridding Options

Each gridding method has its own set of gridding options. Some of the options are the same or similar for the different gridding methods, while other options are specific to particular gridding methods. The following options are available to multiple gridding methods.

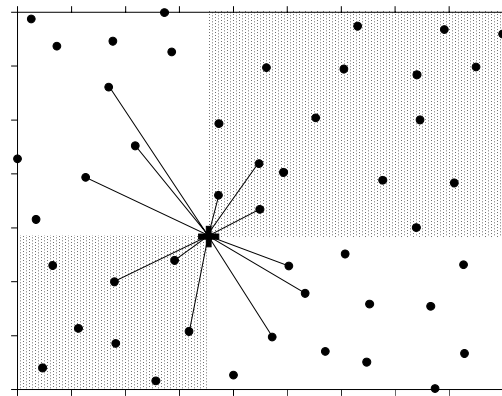
Search

Search options control which data points are considered by the gridding operation when interpolating grid nodes. To access the search options, if available with the gridding method, click the *Advanced Options* button in the **Grid Data** dialog. If searching is available with the selected gridding method, a **Search** page appears in the advanced options dialog. However, in some cases a few of the search options, such as *Anisotropy*, are available on the gridding method's **General** page. If search options are not available for a gridding method, this means that all the data points from the data file must be used when calculating the grid.

No Search

The search options tell **Surfer** how to find data points during the calculation of grid node values. The *No Search (use all of the data)* option tells **Surfer** to use all data when interpolating each grid node. Uncheck the *No Search* box to activate the search during the gridding process.

For small data sets (up to 250 points) the *No Search* option is usually the most appropriate. This type of search increases gridding speed. *No Search* uses all data points in the calculation of every grid node. The distance weighting factors are still applied. Therefore, although a point far removed from the grid node is still used when calculating the grid node value, it carries relatively little weight compared to data points close to the grid node.



This is an example of a four-sector search. The three nearest points in each quadrant are included during gridding in this example.

When data points are evenly distributed over the map area, the *No Search* option is adequate. When observations are heavily clustered within the map area, a four-sector or eight-sector search is recommended. These types of searches are also appropriate when you have data collected on widely spaced traverses. A one-sector search might attempt to estimate grid nodes using data points from a single direction. This might generate unrealistic slopes between traverses, and unrealistic polygonal shaped plateaus across the map area. Four- or eight- sector searches should eliminate or reduce this effect.

Some gridding methods construct an internal matrix based on the number of search points (such as kriging and radial basis function). This matrix can consume a large amount of memory if too many search points are used. For these methods, the default cut off for using all data (*No Search*) versus searching is 250 data points. If there are more than 250 points, **Surfer** defaults to performing a search. If there are 250 or fewer data points, **Surfer** selects the *No Search (use all of the data)* option by default. The absolute maximum number of data points that can be used with the *No Search (use all of the data)* option with these methods is 750. If you have more than 750 points, the *No Search (use all of the data)* option is disabled.

Search Rules

Search rules define the number of points included in interpolating a grid node value. Uncheck the *No Search (use all of the data)* option to enable the search rules edit controls. Search rules limit the number of data points to include in the interpolation at each grid node. Search rules work in concert with the search ellipse. The search ellipse specifies the size of the local neighborhood in which to look for data, and the search rules specify the number of points to actually consider within the neighborhood.

If the number of data points defined by the *Minimum number of data* in all sectors is not found within the *Search Ellipse* distance, the blanking value is assigned at the grid node. Blanking values indicate that insufficient data existed to satisfy the search criteria at that particular location. Blanked grid nodes truncate contour lines on contour maps, and produce low flat regions on wireframes.

Search Options

- The *No Search (use all of the data)* option tells **Surfer** to use all data when interpolating each grid node. Uncheck the *No Search* box to activate the search during the gridding process.
- The *Number of sectors to search* option divides the search area into smaller sections to which you can apply the following three search rules. You can specify up to 32 search sectors.
- The *Maximum number of data to use from ALL sectors* value limits the total number of points used when interpolating a grid node.
- The *Maximum number of data to use from EACH sector* value specifies the number of points to be used from each sector.
- The *Minimum number of data in all sectors (node is blanked if fewer)* value assures that the specified number of points is encountered when interpolating a grid node. If the minimum number of points is not found, the blanking value is assigned at the grid node. Data points beyond the nearest points in a quadrant are ignored even if the data points in another quadrant are farther from the grid node being calculated.
- *Blank node if more than this many sectors are empty* assures that if more empty sectors than this value are encountered, the blanking value is assigned at the grid node.

Chapter 4 - Creating Grid Files

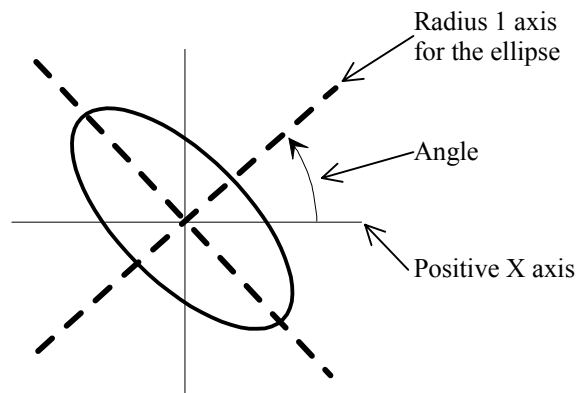
Search Ellipse

The *Search Ellipse* defines the local neighborhood of points to consider when interpolating each grid node. This defines the distance in data units from the grid node that **Surfer** looks to find data points when calculating grid nodes. Data points outside the search ellipse are not considered during grid node interpolation.

Elliptical searches do not impart extra weight to data points in the various directions, but do search farther along one ellipse axis. The default *Search Ellipse* is circular, meaning that **Surfer** looks the same distance in all directions.

Search ellipses are specified by defining the ellipse radii and the angle for the ellipse.

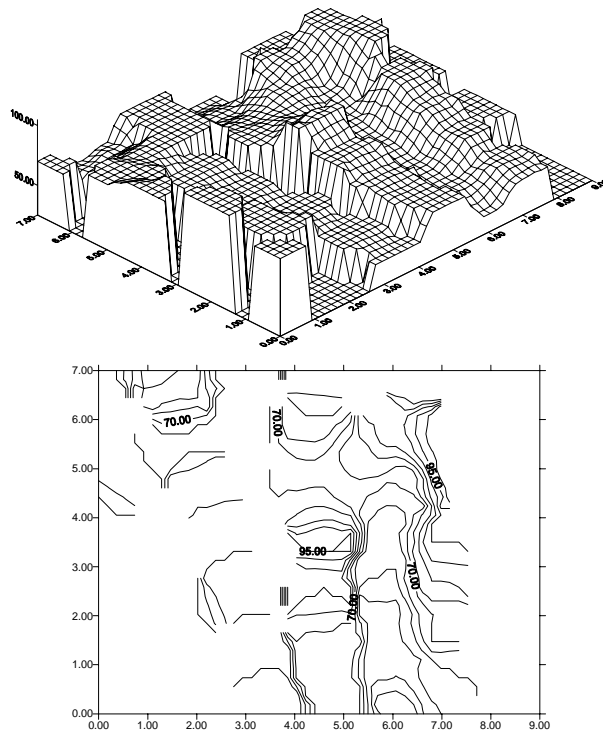
- *Radius 1* and *Radius 2* are positive values indicating the distance in data units.
- *Angle* is the inclination between the positive X axis and the ellipse axis associated with Radius 1. This can be any value between -360 and +360 degrees.



The search ellipse angle is the angle between the positive X axis and the ellipse axis associated with Radius 1.

Blanking values indicate insufficient data existed to generate a grid node value at that particular location based on the specified search rules. For example, if you inadvertently set your search ellipse size to be smaller than half the distance between your data points, a significant number of grid nodes may be blanked in the grid file. Blanked grid nodes cause truncation of contour lines within contour maps, and flat regions on wireframes. In addition, blanking values are placed outside of the convex hull of the data for non-extrapolating gridding methods, like triangulation with linear interpolation and natural neighbor.

Although not required, in most cases it works well to set the search ellipse ratio and direction to coincide with the anisotropy ratio and direction. For more information on anisotropy, see the *Anisotropy* section on page 108.

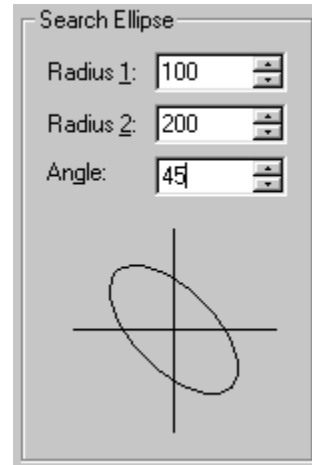


When a search ellipse is set too small, the blanking value is assigned to the grid nodes where not enough data were found within the search ellipse. This can result in an incomplete contour map, with truncation of contour lines, or a wireframe with pronounced discontinuities.

Chapter 4 - Creating Grid Files

To specify the search options:

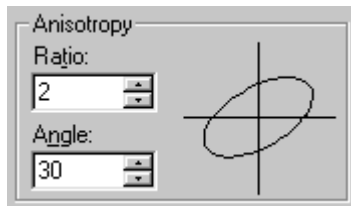
1. Select **Grid | Data** and choose a data file.
2. Select a *Gridding Method*.
3. Click the *Advanced Options* button.
4. Click the **Search** page in the gridding method's advanced options dialog. If the search page does not exist, a sector search is not allowed (the gridding method uses all of the data).
5. If the *No Search (use all of the data)* option is checked, click in the box to deselect this option if you want to set search rules.
6. Specify the *Number of sectors to search*. The number of sectors can range from one to 32.
7. Specify the search rules so the desired number of points is considered during interpolation at each grid node.
8. Create the search ellipse by defining the radius and angle values. The *Search Ellipse* group presents a sketch of the ellipse as specified.
9. Click OK to return to the **Grid Data** dialog.
10. Click OK to grid the data.



The search ellipse graphic reflects changes made to search ellipse options.

Anisotropy

Natural phenomena are created by physical processes. Often these physical processes have preferred orientations. For example, at the mouth of a river the coarse material settles out fastest, while the finer material takes longer to settle. Thus, the closer one is to the shoreline the coarser the sediments while the further from the shoreline the finer the sediments. When interpolating at a point, an observation 100 meters away but in a direction parallel to the shoreline is more likely to be similar to the value at the interpolation point than is an equidistant observation in a direction perpendicular to the shoreline. Anisotropy takes these trends in the data into account during the gridding process.



The anisotropy settings have an axis ratio and angle.

Usually, points closer to the grid node are given more weight than points farther from the grid node. If, as in the example above, the points in one direction have more similarity than points in another direction, it is advantageous to give points in a specific direction more weight in determining the value of a grid node. The relative weighting is defined by the anisotropy ratio.

The underlying physical process producing the data, as well as the sample spacing of the data, are important in the decision of whether or not to reset the default anisotropy settings.

Anisotropy is also useful when data sets use fundamentally different units in the X and Y dimensions. For example, consider plotting a flood profile along a river. The X coordinates are locations, measured in miles along the river channel. The Y coordinates are time, measured in days. The Z values are river depth as a function of location and time. Clearly in this case, the X and Y coordinates would not be plotted on a common scale, because one is distance and the other is time. One unit of X does not equal one unit of Y. While the resulting map can be displayed with changes in scaling, it may be beneficial to apply anisotropy as well.

Another example of when anisotropy might be employed is an isotherm map (contour map) of average daily temperature over the upper Midwest. Although the X and Y coordinates (Easting and Northing) are measured using the same units, along the east-west lines (X lines) the temperature tends to be very similar. Along north-south lines (Y lines) the temperature tends to change more quickly (getting colder as you head north). When gridding the data, it would be advantageous to give more weight to data along the east-west axis than along the north-south axis. When interpolating a grid node, observations that lie in an east-west direction are given greater weight than observations lying an equivalent distance in the north-south direction.

The anisotropy settings include a *Ratio* and an *Angle* setting. The *Ratio* is the maximum range divided by the minimum range. An anisotropy ratio less than two is considered mild, while an anisotropy ratio greater than four is considered severe. Typically, when the anisotropy ratio is greater than three its effect is clearly visible on grid-based maps. The *Angle* is the preferred orientation (direction) of the major axis in degrees.

Chapter 4 - Creating Grid Files

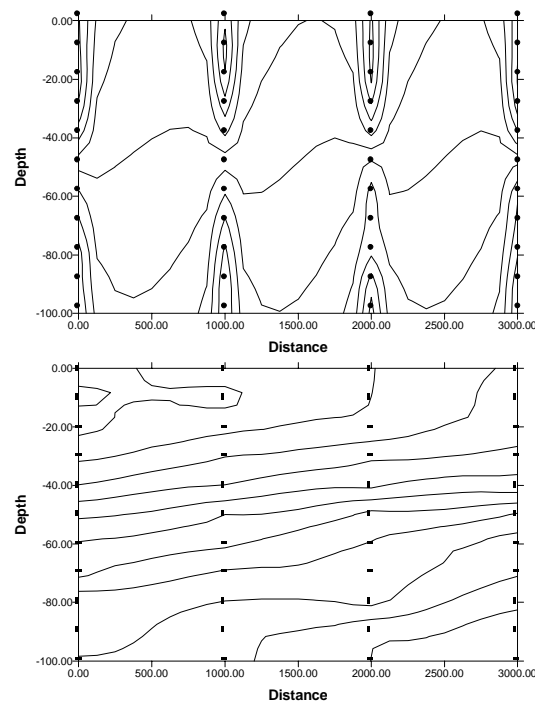
Example

An example where an anisotropy ratio is appropriate is an oceanographic survey to determine water temperature at varying depths. Assume the data are collected every 1000 meters along a survey line, and temperatures are taken every ten meters in depth at each sample location. With this type of data set in mind, consider the problem of creating a grid file. When computing the weights to assign to the data points, closer data points get greater weights than points farther away. A temperature at 10 meters in depth at one location is similar to a sample at 10 meters in depth at another location, although the sample locations are thousands of meters apart. Temperatures might vary greatly with depth, but not as much between sample locations.

NOTE: Unless there is a good reason to use an anisotropy ratio, you should accept the default value of 1.0.

Detailed discussions of anisotropy and kriging as well as anisotropy equations are given in *Chapter 5, Variograms*.

For each different gridding method, anisotropy might be specified in a slightly different manner.



In the oceanographic survey described here, the contour lines cluster around the data points when an anisotropy ratio is not employed. In the bottom contour map, an anisotropy ratio results in contour lines that are a more accurate representation of the data.

Breaklines and Faults

Breaklines and faults are a means to show discontinuities in the surface. Select gridding methods support breaklines and/or faults. Breaklines and faults are defined with blanking files.

If your grid is not dense enough, the breakline or fault will not show very well in the map. If you cannot see any indication of the breakline or fault (i.e. contours do not bend properly), regrid the data with a denser grid.

To include the breakline or fault as a line on your map, use **Map | Base Map** and choose your blanking file. Select all maps and use **Map | Overlay Maps** to position the line correctly. Contour maps have a *Fault Line Properties* button, so you do not need to use a base map to show faults on contour maps.

Defining Breaklines and Faults with Blanking Files

You can create a blanking file to define a breakline or fault in the **Surfer** worksheet or any text editor. These blanking files do not require the blank inside or blank outside flag. Blanking files are described in detail in *Appendix C, File Formats*.

Breaklines

Enter a header containing the number of vertices in the breakline, followed by the XYZ coordinates of each vertex, one per line.

```

3 ← header
2.97 8.32 10
XYZ vertices → 3.38 7.25 20
3.04 5.31 30
  
```

A breakline uses Z values in a blanking file.

Faults

Enter a header containing the number of vertices in the fault, followed by the XY coordinates of each vertex, one per line.

```

3 ← header
2.97 8.32
XY vertices → 3.38 7.25
3.04 5.31
  
```

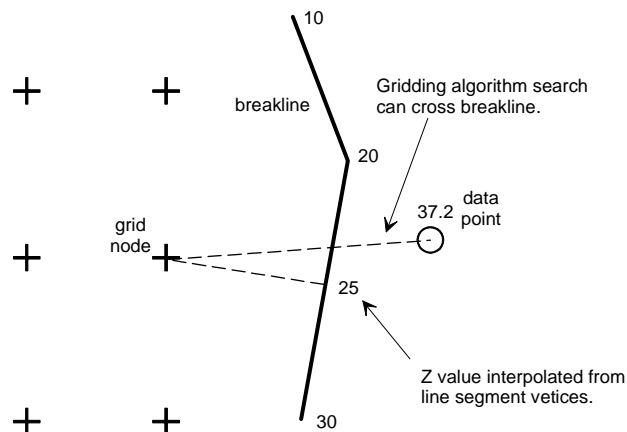
A fault is defined by a blanking file.

Grid Files

The only grid file format that retains fault information is the *Surfer 7* grid file. If you use another grid file format, the faulting information is lost.

Breaklines

A breakline is a three-dimensional blanking file that defines a line with X, Y, and Z values at each vertex. When the gridding algorithm sees a breakline, it calculates the Z value of the nearest point along the breakline, and uses that value in combination with nearby data points to calculate the grid node value. **Surfer** uses linear interpolation to determine the values between breakline vertices when gridding. Unlike faults, breaklines are not barriers to information flow, and the gridding algorithm can cross the breakline to use a point on the other side of the breakline. If a point lies on the breakline, the value of the breakline takes precedence over the point. Breakline applications include defining streamlines, ridges, and other breaks in the slope.



The gridding algorithm search uses a single Z value at the closest point along the line. The search can cross the breakline to search for data.

Gridding algorithm searches use the specified anisotropy ratio when determining the distance to the breakline. Breaklines are ignored when determining this distance. The use of complex breaklines or a large number of breaklines will slow the gridding process significantly. Breaklines cannot cross other breaklines or faults. Breaklines are not allowed with the *No Search (use all of the data)* search method for any gridding method.

The following gridding methods support breaklines:

Inverse Distance to a Power

Radial Basis Function

Kriging

Moving Average

Minimum Curvature

Data Metrics

Nearest Neighbor

Local Polynomial

Faults

In **Surfer**, a fault is a two-dimensional blanking file defining a line acting as a barrier to information flow when gridding. When gridding a data set, data on one side of a fault is not directly used when calculating grid node values on the other side of the fault.

If the fault line is a closed polygon, the gridding algorithm grids data in the interior of the polygon. If the fault line is not a closed polygon, the gridding algorithm can search around the end of the fault to see a point on the other side of the fault, but this longer distance reduces the weight of the point in interpolating the grid node value. If a point lies directly on the fault line, random round-off error determines which side of the fault captures the point.

The following gridding methods support faults:

Inverse Distance to a Power

Minimum Curvature

Nearest Neighbor


Data Metrics

Faults and Memory

Faults consume memory and increase gridding time in proportion to the square of the number of fault segments. In addition, they may cause some gridding methods to fail. With minimum curvature, the failure is due to lack of convergence.

Gridding with Breaklines and Faults

To use a breakline or fault in gridding:

1. Create a blanking file containing the breakline or fault trace. Remember that breaklines need Z values in the blanking file.
2. Click on **Grid | Data**.
3. Choose a data file in the **Open** dialog.
4. In the **Grid Data** dialog, choose a *Gridding Method* that supports breaklines or faults.
5. Click the *Advanced Options* button.
6. Click the **Breaklines** page or **Breaklines and Faults** page, depending on the selected gridding method.
7. Click the *File Containing Breaklines* or *File Containing Fault Traces*  button to select the blanking file [.BLN] containing the faults or breaklines. The number of traces and the number of total vertices are displayed under the file name.
8. Click OK to return to the **Grid Data** dialog.
9. Click OK in the **Grid Data** dialog to grid the data with a breakline or fault.

Gridding Methods

Gridding methods are selected in the **Grid Data** dialog, which is accessed through **Grid | Data**. When you select a gridding method in this dialog, you can specify the parameters for the particular method by clicking on the *Advanced Options* button.

The differences between gridding methods are in the mathematical algorithms used to compute the weights during grid node interpolation. Each method can result in a different representation of your data. It is advantageous to test each method with a typical data set to determine the gridding method that provides you with the most satisfying interpretation of your data.

Inverse Distance to a Power

The *Inverse Distance to a Power* gridding method is a weighted average interpolator, and can be either an exact or a smoothing interpolator.

With inverse distance to a power, data are weighted during interpolation such that the influence of one point relative to another declines with distance from the grid node. Weighting is assigned to data using a weighting power that controls how the weighting factors drop off as distance from a grid node increases. The greater the weighting power, the less effect points far from the grid node have during interpolation. As the power increases, the grid node value approaches the value of the nearest point. For a smaller power, the weights are more evenly distributed among the neighboring data points.

Normally, inverse distance to a power behaves as an exact interpolator. When calculating a grid node, the weights assigned to the data points are fractions, and the sum of all the weights is equal to 1.0. When a particular observation is coincident with a grid node, the distance between that observation and the grid node is 0.0, and that observation is given a weight of 1.0, while all other observations are given weights of 0.0. Thus, the grid node is assigned the value of the coincident observation. The *Smoothing* parameter is a mechanism for buffering this behavior. When you assign a non-zero *Smoothing* parameter, no point is given an overwhelming weight so that no point is given a weighting factor equal to 1.0.

One of the characteristics of inverse distance to a power is the generation of "bull's-eyes" surrounding the position of observations within the gridded area. You can assign a smoothing parameter during inverse distance to a power to reduce the "bull's-eye" effect by smoothing the interpolated grid.

Inverse distance to a power is a very fast method for gridding. With less than 500 points, you can use the *No Search (use all of the data)* search type and gridding proceeds rapidly.

The equation used for *Inverse Distance to a Power* is:

$$\hat{Z}_j = \frac{\sum_{i=1}^n \frac{Z_i}{h_{ij}^\beta}}{\sum_{i=1}^n \frac{1}{h_{ij}^\beta}}$$

$$h_{ij} = \sqrt{d_{ij}^2 + \delta^2}$$

where

h_{ij} is the effective separation distance between grid node j and the neighboring point i

\hat{Z}_j is the interpolated value for grid node j

Z_i are the neighboring points

d_{ij} is the distance between the grid node j and the neighboring point i

β is the weighting power (the *Power* parameter)

δ is the *Smoothing* parameter

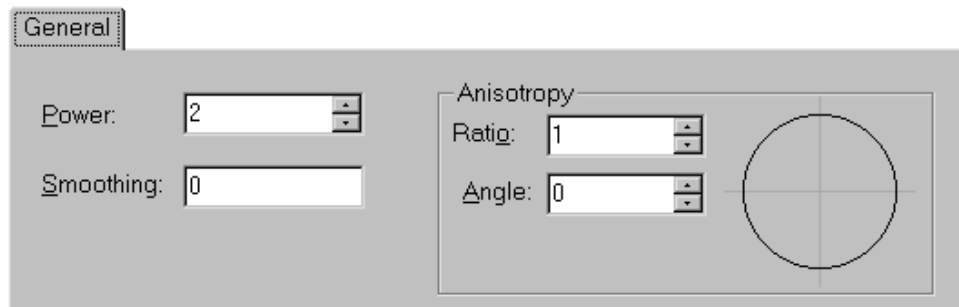
Inverse Distance to a Power Advanced Options

To set inverse distance to a power options:

1. Click on **Grid | Data**.
2. Open a data file.
3. In the **Grid Data** dialog, choose *Inverse Distance to a Power* in the *Gridding Method* group.
4. Click the *Advanced Options* button to display the **Inverse Distance Advanced Options** dialog.
5. On the **General** page, specify the weighting *Power* and *Smoothing* parameters to apply during the gridding operation.
 - The weighting *Power* parameter determines how quickly weights fall off with distance from the grid node. As the power parameter approaches zero, the generated surface approaches a horizontal planar surface through the average of all observations from the data file. As the power parameter increases, the generated surface is a "nearest neighbor" interpolator and the resultant surface becomes polygonal. The polygons represent the

Chapter 4 - Creating Grid Files

- nearest observation to the interpolated grid node. Power values between $1.2\text{e-}038$ and $1.0\text{e+}038$ are accepted, although powers should usually fall between one and three.
- The *Smoothing* factor parameter allows you to incorporate an "uncertainty" factor associated with your input data. The larger the smoothing factor parameter, the less overwhelming influence any particular observation has in computing a neighboring grid node.
6. You can set *Anisotropy* parameters on the **General** page. For more information about anisotropy options, see *Anisotropy* beginning on page 108.
 8. The **Search** page allows you to specify search rules. For more information on search, see the *Search* section beginning on page 103.
 9. The **Breaklines and Faults** page is used to add breaklines and faults to the gridding process. For more information on breaklines and faults, see the *Breaklines and Faults* section beginning on page 111.
 10. Click OK in the **Inverse Distance Advanced Options** dialog to return to the **Grid Data** dialog.
 11. Click OK in the **Grid Data** dialog and a grid file is produced using the **Inverse Distance to a Power** gridding method with the specified options.



The **General** page in the **Inverse Distance Advanced Options** dialog allows you to specify the parameters to use when gridding with inverse distance to a power.

Inverse Distance to a Power References

Davis, John C. (1986), *Statistics and Data Analysis in Geology*, John Wiley and Sons, New York.

Franke, R. (1982), Scattered Data Interpolation: Test of Some Methods, *Mathematics of Computations*, v. 33, n. 157, p. 181-200.


Kriging

Kriging is a geostatistical gridding method that has proven useful and popular in many fields. This method produces visually appealing maps from irregularly spaced data. Kriging attempts to express trends suggested in your data, so that, for example, high points might be connected along a ridge rather than isolated by bull's-eye type contours.

Kriging is a very flexible gridding method. You can accept the kriging defaults to produce an accurate grid of your data, or kriging can be custom-fit to a data set by specifying the appropriate variogram model. Within **Surfer**, kriging can be either an exact or a smoothing interpolator depending on the user-specified parameters. It incorporates anisotropy and underlying trends in an efficient and natural manner.

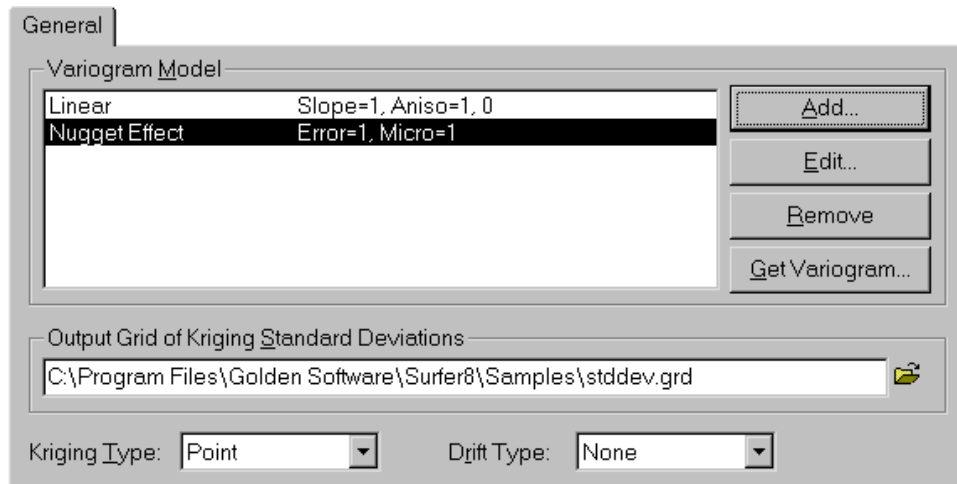
Kriging Advanced Options

To set advanced kriging options:

1. Click on **Grid | Data**.
2. In the **Open** dialog, select a data file and then click the *Open* button.
3. In the **Grid Data** dialog, choose *Kriging* in the *Gridding Method* group.
4. Click the *Advanced Options* button to display the **Kriging Advanced Options** dialog.
5. On the **General** page:
 - Click the *Add* button to add variogram components. Refer to the *Variograms* section for more information on the variogram components.
 - Click the *Edit* button to display the **Variogram Components** dialog. You can edit various parts of the variogram model from this dialog. Once you are finished editing the components, click OK to return to the **Kriging Advanced Options** dialog.
 - To remove a component from the variogram model, highlight the component in the *Variogram Model* group, and then click the *Remove* button.
 - If you have modeled your data with **Surfer's** variogram modeling feature (**Grid | New Variogram**), click the *Get Variogram* button to use the variogram results. The *Get Variogram* button extracts the current variogram model and inserts it into the kriging variogram model. You must have the variogram open in the current plot window to use the *Get Variogram* option.
 - Click the  button to enter a file name into the *Output Grid of Kriging Standard Deviations* box. See *Kriging Standard Deviations* for more information on this subject.
 - Choose *Point* or *Block Kriging* from the *Kriging Type* box. For more information on *Point* or *Block Kriging*, see the *Kriging Type* section.

Chapter 4 - Creating Grid Files

- You can select a *Linear* or *Quadratic* drift type. Drift type *None* is ordinary kriging, while *Linear* and *Quadratic* drift types are universal kriging.
6. The **Search** page allows you to specify search rules. For more information on search, see the *Search* section beginning on page 103.
 7. The **Breaklines** page is used to add breaklines to the gridding process. (Faults are no longer supported with kriging.) For more information on breaklines, see the *Breaklines and Faults* section beginning on page 111.
 8. Click OK in the **Kriging Advanced Options** dialog to return to the **Grid Data** dialog.
 9. Click OK in the **Grid Data** dialog and a grid file is produced using the kriging gridding method with the specified options.



*You can specify the variogram parameters, kriging standard deviations file, kriging type, and drift type in the **Kriging Advanced Options** dialog.*

Variograms

An important component of the kriging gridding method is the variogram. Variograms are discussed in detail in *Chapter 5, Variograms*. The default linear variogram provides a reasonable grid in most circumstances, as long as a nugget effect is not used. A variogram-modeling feature is included in **Surfer** to assist you in selecting the appropriate variogram model for your data. Do not change the variogram components without modeling the variogram first. If you do not understand variograms or variogram modeling, use the default linear variogram with no nugget effect. When in doubt, use the default linear variogram.

Kriging Standard Deviations

Enter a path and file name into the *Output Grid of Kriging Standard Deviations* box to produce an estimation standard deviation grid. If this box is empty, the estimation standard deviation grid is not created.

The kriging standard deviation grid output option greatly slows the kriging process. This is contrary to what you may expect since the kriging variances are usually a by-product of the kriging calculations. However, **Surfer** uses a highly optimized algorithm for calculating the node values. When the variances are requested, a more traditional method must be used, which takes much longer.

There are several cases where a standard deviation grid is incorrect or meaningless. If the variogram model is not truly representative of the data, the standard deviation grid is not helpful to your data analysis. In addition, the kriging standard deviation grid generated when using a variogram model estimated with the *Standardized Variogram* estimator or the *Autocorrelation* estimator is not correct. These two variogram estimators generate dimensionless variograms, so the kriging standard deviation grids are incorrectly scaled. Similarly, while the default linear variogram model will generate useful contour plots of the data, the associated kriging standard deviation grid is incorrectly scaled and should not be used. The default linear model slope is one, and since the kriging standard deviation grid is a function of slope, the resulting grid is meaningless.

The kriging standard deviation grid is not allowed with the *No Search (use all of the data)* option on the **Search** page.

Kriging Type

Surfer includes two kriging types: *Point Kriging* and *Block Kriging*. A detailed discussion of the two methods can be found in Isaaks and Srivastava (1989, Chapters 12 and 13). *Ordinary* (no drift) and *Universal Kriging* (linear or quadratic drift) algorithms can be applied to both kriging types.

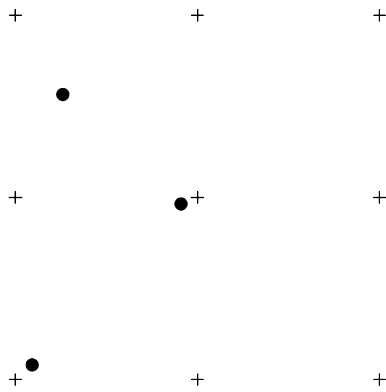
Both *Point Kriging* and *Block Kriging* generate an interpolated grid. *Point Kriging* estimates the values of the points at the grid nodes. *Block Kriging* estimates the average value of the rectangular blocks centered on the grid nodes. The blocks are the size and shape of a grid cell. Since *Block Kriging* is estimating the average value of a block, it generates smoother contours. Furthermore, since *Block Kriging* is not estimating the value at a point, *Block Kriging* is not a perfect interpolator. That is even if an observation falls exactly on a grid node, the *Block Kriging* estimate for that node does not exactly reproduce the observed value.

Chapter 4 - Creating Grid Files

When a *Kriging* standard deviation grid is generated with *Block Kriging*, the generated grid contains the *Block Kriging* standard deviations and not the *Point Kriging* standard deviations.

The numerical integration required for point-to-block variogram calculations necessary for *Block Kriging* is carried out using a 3x3, two-dimensional Gaussian-Quadrature. In particular, **Surfer** uses Section 25.4.62 of Abramowitz and Stegun (1972).

Point Kriging is the default method.



In the diagram to the left, the crosses indicate a block of grid nodes and the filled circles indicate data points. If we were to interpolate the center grid node with Point Kriging, the data point closest to the center grid node would have the greatest weight in determining the value of the grid node. If we were to interpolate the center grid node with Block Kriging, all three data points within the block of grid nodes are averaged to interpolate the grid node value.

Kriging References

For a detailed derivation and discussion of kriging, see Cressie (1991) or Journel and Huijbregts (1978). Journel (1989) is, in particular, a concise presentation of geostatistics (and kriging). Isaaks and Srivastava (1989) offer a clear introduction to the topic, though it does not cover some of the more advanced details. For those who need to see computer code to really understand an algorithm, Deutsch and Journel (1992) includes a complete, well-written, and well-documented source code library of geostatistics computer programs (in FORTRAN). Finally, a well-researched account of the history and origins of kriging can be found in Cressie (1990).

Abramowitz, M., and Stegun, I. (1972), *Handbook of Mathematical Functions*, Dover Publications, New York.

Cressie, N. A. C. (1990), The Origins of Kriging, *Mathematical Geology*, v. 22, p. 239-252.

Cressie, N. A. C. (1991), *Statistics for Spatial Data*, John Wiley and Sons, Inc., New York, 900 pp.

Deutsch, C.V., and Journel, A. G. (1992), *GSLIB - Geostatistical Software Library and User's Guide*, Oxford University Press, New York, 338 pp.

Isaaks, E. H., and Srivastava, R. M. (1989), *An Introduction to Applied Geostatistics*, Oxford University Press, New York, 561 pp.

Journel, A.G., and Huijbregts, C. (1978), *Mining Geostatistics*, Academic Press, 600 pp.

Journel, A.G. (1989), *Fundamentals of Geostatistics in Five Lessons*, American Geophysical Union, Washington D.C.

Minimum Curvature

Minimum Curvature is widely used in the earth sciences. The interpolated surface generated by minimum curvature is analogous to a thin, linearly elastic plate passing through each of the data values with a minimum amount of bending. Minimum curvature generates the smoothest possible surface while attempting to honor your data as closely as possible. Minimum curvature is not an exact interpolator, however. This means that your data are not always honored exactly.

Minimum curvature produces a grid by repeatedly applying an equation over the grid in an attempt to smooth the grid. Each pass over the grid is counted as one iteration. The grid node values are recalculated until successive changes in the values are less than the *Maximum Residual* value, or the maximum number of iterations is reached.

Surfer fully implements the concepts of tension as described and detailed in Smith and Wessel (1990). Also, as recommended by Smith and Wessel (1991), this routine first fits a simple planar model using least square regression:

$$AX + BY + C = Z(X, Y)$$

There are four steps to generate the final grid using the minimum curvature method.

1. The least square regression model is fit to the data.
2. The values of the planar regression model at the data locations are subtracted from the data values; this yields a set of residual data values.
3. The minimum curvature algorithm is used to interpolate the residuals at the grid nodes.
4. The values of the planar regression model at the grid nodes are added to the interpolated residuals, yielding a final interpolated surface.

Chapter 4 - Creating Grid Files

Unlike Smith and Wessel (1990), the fixed nodes are defined as the average of the neighboring observed values. That is, consider a rectangle the size and shape of a grid cell. The neighborhood of a grid node is defined by this rectangle centered on the grid node. If there are any observed data within the neighborhood of a grid node, the value of that grid node is fixed equal to the arithmetic average of contained data.

The minimum curvature algorithm generates the surface that interpolates the available data and solves the modified biharmonic differential equation with tension:

$$(1 - T_i) \nabla^2 (\nabla^2 Z) - (T_i) \nabla^2 Z = 0$$

There are three sets of associated boundary conditions:

On the edges:
$$(1 - T_b) \frac{\partial^2 Z}{\partial n^2} + (T_b) \frac{\partial Z}{\partial n} = 0$$

On the edges:
$$\frac{\partial (\nabla^2 Z)}{\partial n} = 0$$

At the corners:
$$\frac{\partial^2 Z}{\partial x \partial y} = 0$$

where

∇^2 is the Laplacian operator

n is the boundary normal

T_i is the internal tension

T_b is the boundary tension

Minimum Curvature Advanced Options

To set minimum curvature advanced options:

1. Click on **Grid | Data**.
2. In the **Open** dialog, select a data file and then click the *Open* button.
3. In the **Grid Data** dialog, choose *Minimum Curvature* in the *Gridding Method* group.
4. Click the *Advanced Options* button to display the **Minimum Curvature Advanced Options** dialog.
5. You can control the convergence criteria for minimum curvature on the **General** page:
 - The *Maximum Residual* parameter has the same units as the data, and an appropriate value is approximately 10% of the data precision. If data values are measured to the nearest 1.0 unit, the *Maximum Residual* value should be set at 0.1. The iterations continue until the maximum grid node correction for the entire iteration is less than the *Maximum Residual* value. The default *Maximum Residual* value is given by: *Default Max Residual = 0.001 (Zmax - Z min)*
 - The *Maximum Iteration* parameter should be set at one to two times the number of grid nodes generated in the grid file. For example, when generating a 50 by 50 grid using *Minimum Curvature*, the *Maximum Iteration* value should be set between 2,500 and 5,000.
 - For more information on *Internal Tension* and *Boundary Tension*, see the *Internal and Boundary Tension* section.
 - For more information on the *Relaxation Factor*, see the *Relaxation Factor* section.
 - You can set *Anisotropy* parameters in this dialog. For more information about anisotropy options refer to the *Anisotropy* section beginning on page 108.

The screenshot shows the 'General' page of the 'Minimum Curvature Advanced Options' dialog. It features a list of input fields for various parameters: 'Maximum Residual' is set to 0.081, 'Maximum Iteration' is 100000, 'Relaxation Factor' is 1, 'Internal Tension' is 0, and 'Boundary Tension' is 0. To the right, there is an 'Anisotropy' section with a 'Ratio' set to 1 and a circular diagram with a crosshair.

Click the *Advanced Options* button in the **Grid Data** dialog to set the minimum curvature options.

Chapter 4 - Creating Grid Files

6. The **Breaklines and Faults** page is used to add breaklines and faults to the gridding process. For more information on breaklines and faults, see the *Breaklines and Faults* section beginning on page 111.
7. Click OK in the **Minimum Curvature Advanced Options** dialog to return to the **Grid Data** dialog.
8. Click OK in the **Grid Data** dialog and a grid file is produced using the minimum curvature gridding method with the specified options.

Internal and Boundary Tension

Qualitatively, the minimum curvature gridding algorithm is attempting to fit a piece of sheet metal through all of the observations without putting any creases or kinks in the surface. Between the fixed observation points, the sheet bows a bit. The *Internal Tension* is used to control the amount of this bowing on the interior: the higher the tension, the less the bowing. For example, a high tension makes areas between observations look like facets of a gemstone. The *Boundary Tension* controls the amount of bowing on the edges. By default, the *Internal Tension* and the *Boundary Tension* are set to zero.

Relaxation Factor

The minimum curvature gridding algorithm is actually solving the specified partial differential equation using a successive over-relaxation algorithm. The interior is updated using a "chessboard" strategy, as discussed in Press, et al. (1988, p. 868). The only difference is that the biharmonic equation must have nine different "colors," rather than just black and white.

The *Relaxation Factor* is as described in Press et al. (1988). In general, the *Relaxation Factor* should not be altered. The default value (1.0) is a good generic value. Roughly, the higher the *Relaxation Factor* (closer to two) the faster the *Minimum Curvature* algorithm converges, but the more likely it will not converge at all. The lower the *Relaxation Factor* (closer to zero) the more likely the *Minimum Curvature* algorithm will converge, but the algorithm is slower. The optimal *Relaxation Factor* is derived through trial and error.

Convergence

As mentioned in Briggs (1974), and strongly recommended in Smith and Wessel (1990), **Surfer** uses a "multiple lattice strategy." It starts with a coarse grid and then incrementally refines the grid until the final density is achieved.

The relaxation approach is a local smoothing process and, consequently, short-wavelength components of *Z* are found quickly. On the other hand, the relaxation process does not propagate the effects of the data constraints to longer wavelengths efficiently.

As recommended by Briggs (1974) and Smith and Wessel (1990), this routine determines convergence by comparing the largest magnitude nodal change in one iteration to the specified tolerance (*Maximum Residual*).

The status of the algorithm is reflected on the status line. For example, *pass 2 of 4, iteration 360 (0.1234 > 0.08)* means there are four levels of grids considered (the fourth is the final grid), and the algorithm is currently working on the second. The algorithm is currently on iteration 360. If the iteration number exceeds the *Maximum Iteration* parameter, the algorithm terminates without generating the grid and provides a failure-to-converge warning. The numbers in the parentheses are the current largest residual and the largest allowed residual. The largest allowed residual equals the *Maximum Residual* parameter on the final pass.

The *Minimum Curvature* method requires at least four data points.

Minimum Curvature References

Briggs, I. C. (1974), Machine Contouring Using Minimum Curvature, *Geophysics*, v. 39, n. 1, p. 39-48.

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. (1988), *Numerical Recipes in C*, Cambridge University Press.

Smith, W. H. F., and Wessel, P. (1990), Gridding with Continuous Curvature Splines in Tension, *Geophysics*, v. 55, n. 3, p. 293-305.

Modified Shepard's Method

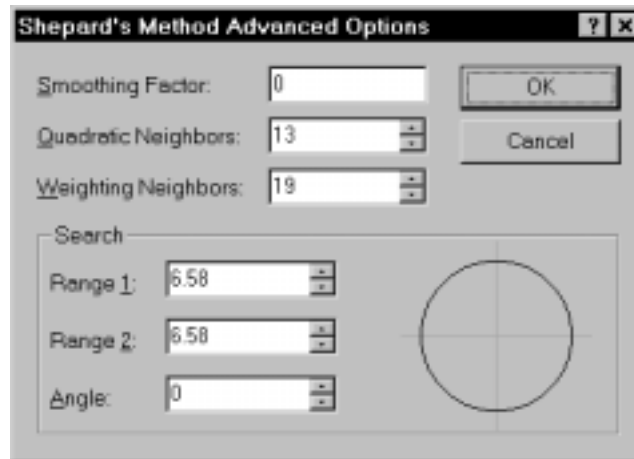
Modified Shepard's Method uses an inverse distance weighted least squares method. As such, modified Shepard's method is similar to the inverse distance to a power interpolator, but the use of local least squares eliminates or reduces the "bull's-eye" appearance of the generated contours. The modified Shepard's method can be either an exact or a smoothing interpolator.

The **Surfer** algorithm implements Franke and Nielson's (1980) modified quadratic Shepard's method with a full sector search as described in Renka (1988).

Modified Shepard's Method Advanced Options

To set the modified Shepard's method options:

1. Click on **Grid | Data**.
2. In the **Open** dialog, select a data file and then click the *Open* button.
3. In the **Grid Data** dialog, choose *Modified Shepard's Method* in the *Gridding Method* group.
4. Click the *Advanced Options* button to display the **Shepard's Method Advanced Options** dialog.
5. You can assign a smoothing parameter to the gridding operation. The *Smoothing Factor* parameter allows *Modified Shepard's Method* to operate as a smoothing interpolator. Greater smoothing occurs as you increase the value of the smoothing parameter. In general, values between zero and one are most reasonable.
6. For more information on *Quadratic Neighbors*, see the *Quadratic Neighbors* section.
7. For more information on *Weighting Neighbors*, see the *Weighting Neighbors* section.
8. You can set *Search* parameters in this dialog. For more information about search options, refer to the *Search* section on page 103.
9. Click OK in the **Shepard's Method Advanced Options** dialog to return to the **Grid Data** dialog.
10. Click OK in the **Grid Data** dialog and a grid file is produced using the *Modified Shepard's Method* gridding method with the specified options.



You can set the modified Shepard's method options, including search, in the **Shepard's Method Advanced Options** dialog.

Quadratic Neighbors

The modified Shepard's method starts by computing a local least squares fit of a quadratic surface around each observation. The *Quadratic Neighbors* parameter specifies the size of the local neighborhood by specifying the number of local neighbors. The local neighborhood is a circle of sufficient radius to include exactly this many neighbors. The default value follows the recommendation of Renka (1988).

Weighting Neighbors

The interpolated values are generated using a distance-weighted average of the previously computed quadratic fits associated with neighboring observations. The *Weighting Neighbors* parameter specifies the size of the local neighborhood by specifying the number of local neighbors. The local neighborhood is a circle of sufficient radius to include exactly this many neighbors. The default value follows the recommendation of Renka (1988).

Modified Shepard's Method References

Franke, R., and Nielson, G. (1980), Smooth Interpolation of Large Sets of Scattered Data, *International Journal for Numerical Methods in Engineering*, v. 15, p. 1691-1704.

Renka, R. J. (1988), Multivariate Interpolation of Large Sets of Scattered Data, *ACM Transaction on Mathematical Software*, v. 14, n. 2, p. 139-148.

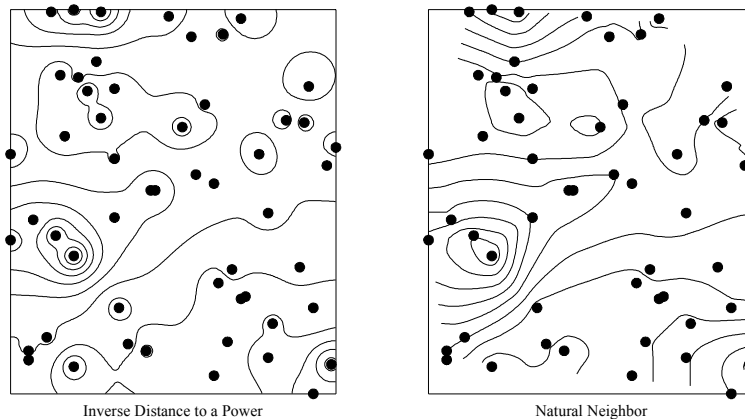
Shepard, D. (1968), A two-dimensional interpolation function for irregularly spaced data, *Proc. 23rd Nat. Conf. ACM*, p. 517-523.

Natural Neighbor

The *Natural Neighbor* gridding method is quite popular in some fields. What is *Natural Neighbor* interpolation? Consider a set of Thiessen polygons (the dual of a Delaunay triangulation). If a new point (target) were added to the data set, these Thiessen polygons would be modified. In fact, some of the polygons would shrink in size, while none would increase in size. The area associated with the target's Thiessen polygon that was taken from an existing polygon is called the "borrowed area." The *Natural Neighbor* interpolation algorithm uses a weighted average of the neighboring observations, where the weights are proportional to the "borrowed area."

Chapter 4 - Creating Grid Files

The *Natural Neighbor* method does not extrapolate contours beyond the convex hull of the data locations (i.e. the outline of the Thiessen polygons).

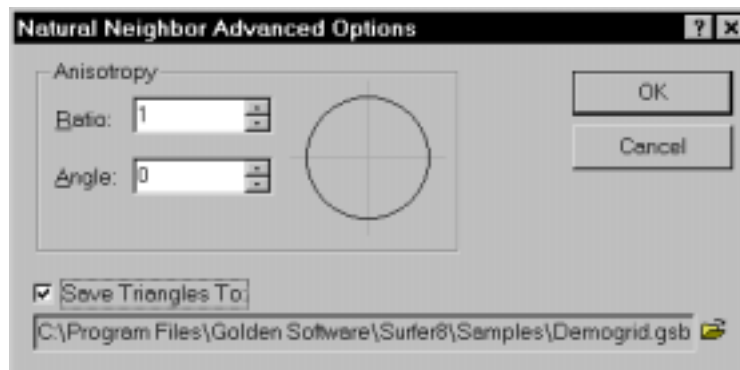


The map on the left shows contours generated by the inverse distance to a power method and the map on the right shows contours generated by the natural neighbor method. Contours are not extrapolated beyond the boundary of the Thiessen polygons with the natural neighbor gridding method.

Natural Neighbor Advanced Options

To set the natural neighbor advanced options:

1. Click on **Grid | Data**.
2. In the **Open** dialog, select a data file and then click the *Open* button.
3. In the **Grid Data** dialog, choose *Natural Neighbor* in the *Gridding Method* group.
4. Click the *Advanced Options* button to display the **Natural Neighbor Advanced Options** dialog.



*You can set the anisotropy options and save Delaunay triangles in the **Natural Neighbor Advanced Options** dialog.*

5. For more information about anisotropy options refer to the *Anisotropy* section beginning on page 108.
6. Check the *Save Triangles To* box to export the Delaunay triangulation in any of the supported export formats. The resulting file can be loaded as a base map and combined with contour maps, surface plots, etc.
7. Click OK in the **Natural Neighbor Advanced Options** dialog to return to the **Grid Data** dialog.
8. Click OK in the **Grid Data** dialog and a grid file is produced using the *Natural Neighbor* gridding method with the specified options.

Natural Neighbor References

The main reference for this method is Sibson (1981), however, also refer to Sibson (1980) and Watson (1994). Watson (1994) discusses the *Natural Neighbor* gridding technique in some detail, though **Surfer** does not use any of Watson's source code.

Sibson, R. (1980), A Vector Identity for the Dirichlet Tessilation, *Math. Proc. Cambridge Phil. Soc.*, v. 87, p. 151-155.

Sibson, R. (1981), A Brief Description of Natural Neighbor Interpolation, *Interpreting Multivariate Data*, V. Barnett editor, John Wiley and Sons, New York, p. 21-36.

Watson, Dave (1994), *Nngridr - An Implementation of Natural Neighbor Interpolation*, David Watson, P.O. Box 734, Claremont, WA 6010, Australia.

Nearest Neighbor

The *Nearest Neighbor* gridding method assigns the value of the nearest point to each grid node. This method is useful when data are already evenly spaced, but need to be converted to a **Surfer** grid file. Alternatively, in cases where the data are nearly on a grid with only a few missing values, this method is effective for filling in the holes in the data.

Sometimes with nearly complete grids of data, there are areas of missing data that you want to exclude from the grid file. In this case, you can set the search to values so that areas of no data are assigned the blanking value in the grid file. By setting the search ellipse radii to values less than the distance between data values in your file, the blanking value is assigned at all grid nodes where data values do not exist.

Chapter 4 - Creating Grid Files

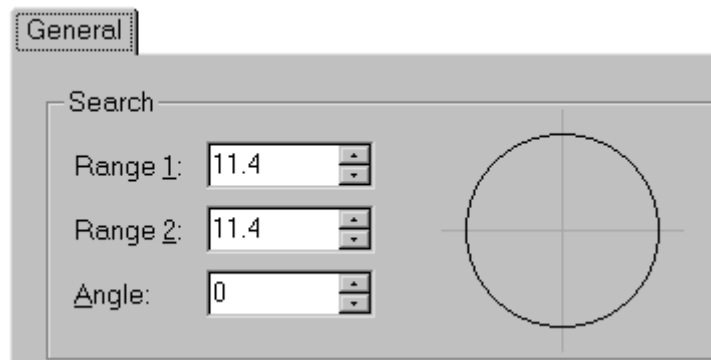
When you use the nearest neighbor method to convert regularly spaced XYZ data to a grid file, you can set the grid spacing equal to the spacing between data points in the file. Refer to *Producing a Grid File from a Regular Array of XYZ Data* for the procedure of converting regularly spaced XYZ data into a **Surfer** grid file.

Breaklines and faults can also be used with the nearest neighbor gridding method.

Nearest Neighbor Advanced Options

To set the nearest neighbor advanced options:

1. Click on **Grid | Data**.
2. In the **Open** dialog, select a data file and then click the *Open* button.
3. In the **Grid Data** dialog, choose *Nearest Neighbor* in the *Gridding Method* group.
4. Click the *Advanced Options* button to display the **Nearest Neighbor Advanced Options** dialog.
5. On the **General** page, set the search range and angle. For more information on search, see the *Search* section beginning on page 103.
6. The **Breaklines and Faults** page is used to add breaklines and faults to the gridding process. For more information on breaklines and faults, see the *Breaklines and Faults* section beginning on page 111.
7. Click OK in the **Nearest Neighbor Advanced Options** dialog to return to the **Grid Data** dialog.
8. Click OK in the **Grid Data** dialog and a grid file is produced using the *Nearest Neighbor* gridding method with the specified options.



*You can set search parameters on the **General** page in the **Nearest Neighbor Advanced Options** dialog.*

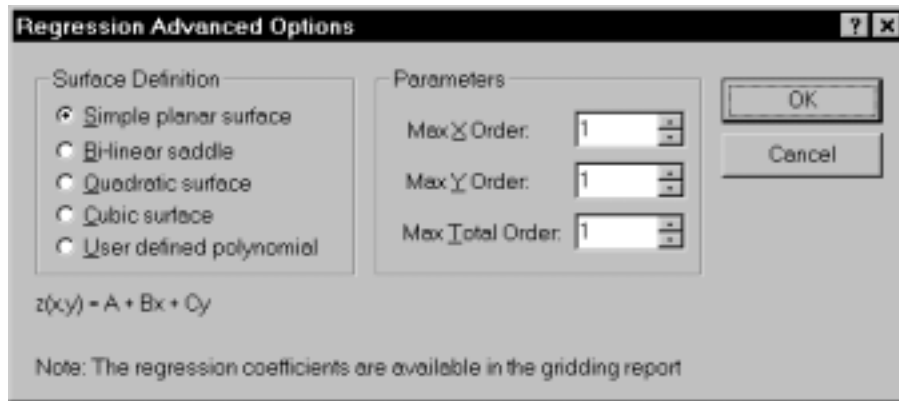
Polynomial Regression

Polynomial Regression is used to define large-scale trends and patterns in your data. Polynomial regression is not really an interpolator because it does not attempt to predict unknown Z values. There are several options you can use to define the type of trend surface.

Regression Advanced Options

To set polynomial regression advanced options:

1. Click on **Grid | Data**.
2. In the **Open** dialog, select a data file and then click the *Open* button.
3. In the **Grid Data** dialog, choose *Polynomial Regression* in the *Gridding Method* group.
4. Click the *Advanced Options* button to display the **Regression Advanced Options** dialog.
5. You can select the type of polynomial regression to apply to your data from the *Surface Definition* group. As you select the different types of polynomials, a generic polynomial form of the equation is presented in the dialog, and the values in the *Parameters* group change to reflect the selection. The available choices are:
 - Simple planar surface*
 - Bi-linear saddle*
 - Quadratic surface*
 - Cubic surface*
 - User defined polynomial*
6. The *Parameters* group allows you to specify the maximum powers for the X and Y component in the polynomial equation. As you change the *Parameters* values, the options are changed in the *Surface Definition* group to reflect the defined parameters.
 - The *Max X Order* specifies the maximum power for the X component in the polynomial equation.
 - The *Max Y Order* specifies the maximum power for the Y component in the polynomial equation.
 - The *Max Total Order* specifies the maximum sum of the *Max X Order* and *Max Y Order* powers. All of the combinations of the X and Y components are included in the polynomial equation as long as the sum of the two powers does not exceed the *Max Total Order* value.
7. Click OK in the **Regression Advanced Options** dialog to return to the **Grid Data** dialog.
8. If you would like a report of the coefficients used in the calculation of the surface, make sure to check the *Generate Report* check box in the **Grid Data** dialog.
9. Click OK in the **Grid Data** dialog and a grid file is produced using the *Polynomial Regression* gridding method with the specified options.



Click the Advanced Options button to open the **Regression Advanced Options** dialog.

Polynomial Regression Reference

Draper, N., and Smith, H. (1981), *Applied Regression Analysis*, second edition, Wiley-Interscience, 709 pp.

Radial Basis Function

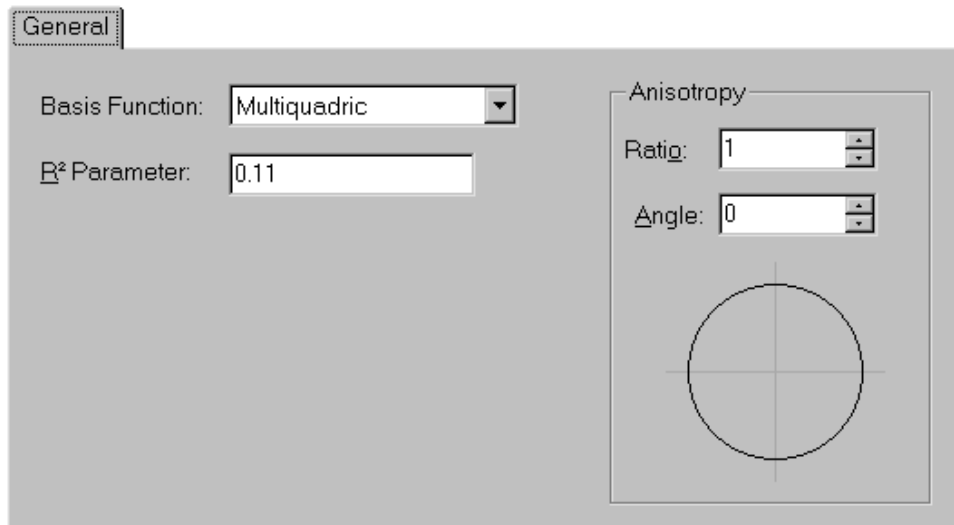
Radial Basis Function interpolation is a diverse group of data interpolation methods. In terms of the ability to fit your data and to produce a smooth surface, the *Multiquadric* method is considered by many to be the best. All of the radial basis function methods are exact interpolators, so they attempt to honor your data. You can introduce a smoothing factor to all the methods in an attempt to produce a smoother surface.

Radial Basis Function Advanced Options

To set radial basis function advanced options:

1. Click on **Grid | Data**.
2. In the **Open** dialog, select a data file and then click the *Open* button.
3. In the **Grid Data** dialog, choose *Radial Basis Function* in the *Gridding Method* group.
4. Click the *Advanced Options* button to display the **Radial Basis Advanced Options** dialog.
5. In the **General** page, you can specify the function parameters for the gridding operation.
 - The *Basis Function* list specifies the basis kernel function to use during gridding. This defines the optimal weights applied to the data points during the interpolation. The *Basis Function* is analogous to the variogram in kriging. Experience indicates that the *Multiquadric* basis function works quite well in most cases. Successful use of the *Thin Plate Spline* basis function is also reported regularly in the technical literature.

- The R^2 *Parameter* is a shaping or smoothing factor. The larger the R^2 *Parameter* shaping factor, the rounder the mountain tops and the smoother the contour lines. There is no universally accepted method for computing an optimal value for this factor. A reasonable trial value for R^2 *Parameter* is between the average sample spacing and one-half the average sample spacing.
6. You can set *Anisotropy* parameters on the **General** page. For more information about anisotropy options see the *Anisotropy* section on page 108.
 7. The **Search** page allows you to specify search rules. For more information on search, see the *Search* section beginning on page 103.
 8. The **Breaklines** page is used to add breaklines to the gridding process. For more information on breaklines, see the *Breaklines and Faults* section beginning on page 111.
 9. Click OK in the **Radial Basis Advanced Options** dialog to return to the **Grid Data** dialog.
 10. Click OK in the **Grid Data** dialog and a grid file is produced using the *Radial Basis Function* gridding method with the specified options.



Set the type of radial basis function and set anisotropy in the **Radial Basis Function Advanced Options** dialog.

Function Types

The basis kernel functions are analogous to variograms in kriging. The basis kernel functions define the optimal set of weights to apply to the data points when interpolating a grid node. The available basis kernel functions are listed in the *Basis Function* list in the **Radial Basis Function Advanced Options** dialog.

Inverse Multiquadric $B(h) = \frac{1}{\sqrt{h^2 + R^2}}$

Multilog $B(h) = \log(h^2 + R^2)$

Multiquadric $B(h) = \sqrt{h^2 + R^2}$

Natural Cubic Spline $B(h) = (h^2 + R^2)^{3/2}$

Thin Plate Spline $B(h) = (h^2 + R^2) \log(h^2 + R^2)$

where

h is the anisotropically rescaled, relative distance from the point to the node

R^2 is the smoothing factor specified by the user

Default R^2 Value

The default value for R^2 in the radial basis function gridding algorithm is calculated as follows:
(length of diagonal of the data extent)² / (25 * number of data points)

Radial Basis Function References

A concise and readable introduction to radial basis function interpolation can be found in Carlson and Foley (1991a). Given the clarity of presentation and the numerous examples, Hardy (1990) provides an excellent overview of the method, although this paper focuses exclusively on the special case of multiquadrics.

Carlson, R.E., and Foley, T.A. (1991a), *Radial Basis Interpolation Methods on Track Data*, Lawrence Livermore National Laboratory, UCRL-JC-1074238.

Carlson, R. E., and Foley, T. A. (1991b), The Parameter R^2 in Multiquadric Interpolation, *Computers Math. Applic.*, v. 21, n. 9, p. 29-42.

Franke, R. (1982), Scattered Data Interpolation: Test of Some Methods, *Mathematics of Computations*, v. 33, n. 157, p. 181-200.

Hardy, R. L. (1990), Theory and Applications of the Multiquadric-BiHarmonic Method, *Computers Math. Applic.*, v. 19, n. 8/9, p. 163-208.

Powell, M.J.D. (1990), *The Theory of Radial Basis Function Approximation in 1990*, University of Cambridge Numerical Analysis Reports, DAMTP 1990/NA11.

Triangulation with Linear Interpolation

The *Triangulation with Linear Interpolation* method uses the optimal Delaunay triangulation. The algorithm creates triangles by drawing lines between data points. The original points are connected in such a way that no triangle edges are intersected by other triangles. The result is a patchwork of triangular faces over the extent of the grid. This method is an exact interpolator.

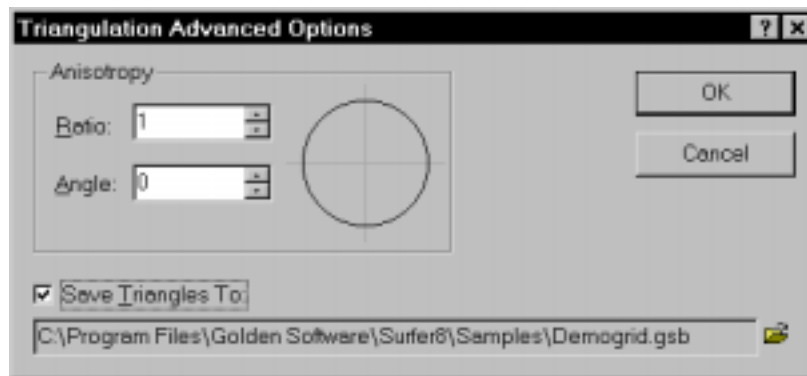
Each triangle defines a plane over the grid nodes lying within the triangle, with the tilt and elevation of the triangle determined by the three original data points defining the triangle. All grid nodes within a given triangle are defined by the triangular surface. Because the original data are used to define the triangles, the data are honored very closely.

Triangulation with linear interpolation works best when your data are evenly distributed over the grid area. Data sets that contain sparse areas result in distinct triangular facets on the map.

Triangulation with Linear Interpolation Advanced Options

To set triangulation with linear interpolation advanced options:

1. Click on **Grid | Data**.
2. In the **Open** dialog, select a data file and then click the *Open* button.
3. In the **Grid Data** dialog, choose *Triangulation with Linear Interpolation* in the *Gridding Method* group.
4. Click the *Advanced Options* button to display the **Triangulation Advanced Options** dialog.
5. For more information about anisotropy options refer to the *Anisotropy* section on page 108.
6. Check the *Save Triangles To* box to export the Delaunay triangulation in any of the supported export formats. The resulting file can be loaded as a base map and combined with contour maps, surface plots, etc.
7. Click OK in the **Triangulation Advanced Options** dialog to return to the **Grid Data** dialog.
8. Click OK in the **Grid Data** dialog and a grid file is produced using the *Triangulation with Linear Interpolation* gridding method with the specified options.



*You can set the anisotropy options and save Delaunay triangles in the **Triangulation Advanced Options** dialog.*

Triangulation with Linear Interpolation References

The **Surfer** implementation of triangulation with linear interpolation is based upon four papers. Lee and Schachter (1980) present a complete discussion of (Delaunay) triangulation, including the details of two algorithms and the underlying mathematical proofs. Lawson (1977) is equally informative. The algorithm presented in Guibas and Stolfi (1985) form the basis for the **Surfer** implementation.

Guibas, L., and J. Stolfi (1985), Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams, *ACM Transactions on Graphics*, v. 4, n. 2, p. 74-123.

Lawson, C. L. (1977), Software for C¹ surface interpolation, *Mathematical Software III*, J. Rice (ed.), Academic Press, New York, p. 161-193.

Lee, D. T., and Schachter, B. J. (1980), Two Algorithms for Constructing a Delaunay Triangulation, *International Journal of Computer and Information Sciences*, v. 9, n. 3, p. 219-242.

Moving Average

The *Moving Average* gridding method assigns values to grid nodes by averaging the data within the grid node's search ellipse.

To use moving average, define a search ellipse and specify the minimum number of data to use. For each grid node, the neighboring data are identified by centering the search ellipse on the node. The output grid node value is set equal to the arithmetic average of the identified neighboring data. If there is fewer than the specified minimum number of data within the neighborhood, the grid node is blanked.

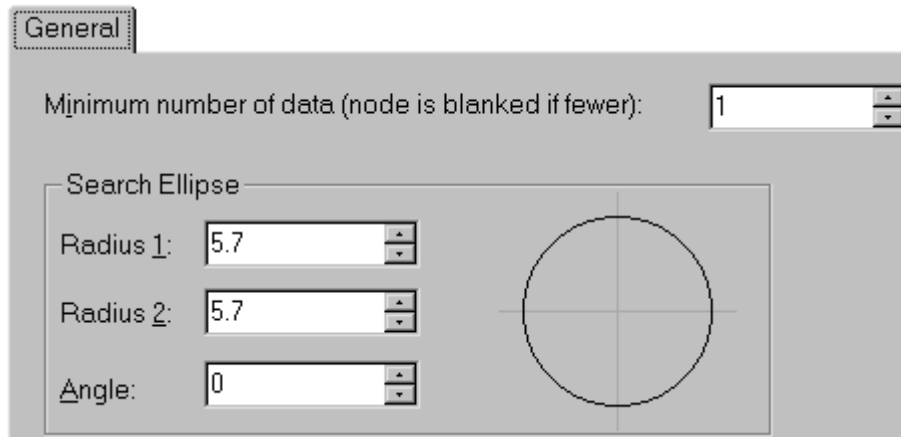
Moving Average Advanced Options

To set moving average advanced options:

1. Click on **Grid | Data**.
2. In the **Open** dialog, select a data file and then click the *Open* button.
3. In the **Grid Data** dialog, choose *Moving Average* in the *Gridding Method* group.
4. Click the *Advanced Options* button to display the **Moving Average Advanced Options** dialog.
5. On the **General** page:

Chapter 4 - Creating Grid Files

- The *Minimum number of data (node is blanked if fewer)* value sets the specified number of points when interpolating a grid node. If the minimum number of points is not found, the blanking value is assigned to the grid node.
- Search ellipses are specified by defining the ellipse radii and the angle for the ellipse. *Radius 1* and *Radius 2* are positive values indicating the distance in data units. *Angle* is the inclination between the positive X axis and the ellipse axis associated with *Radius 1*. This can be any value between -360 and +360 degrees. See the *Search* section on page 103 for more information on search.



The screenshot shows a software dialog box with a 'General' tab. At the top, there is a label 'Minimum number of data (node is blanked if fewer):' followed by a text input field containing the number '1'. Below this, there is a section titled 'Search Ellipse'. Inside this section, there are three input fields: 'Radius 1:' with the value '5.7', 'Radius 2:' with the value '5.7', and 'Angle:' with the value '0'. To the right of these input fields is a graphical representation of a circle (representing an ellipse) with a crosshair in the center, indicating the search area.

You can set the blanking options and the search ellipse in the **Moving Average Advanced Options** dialog.

6. The **Breaklines** page is used to add breaklines to the gridding process. For more information on breaklines, see the *Breaklines and Faults* section beginning on page 111.
7. Click OK in the **Moving Average Advanced Options** dialog to return to the **Grid Data** dialog.
8. Click OK in the **Grid Data** dialog and a grid file is produced using the *Moving Average* gridding method with the specified options.

Data Metrics

The collection of data metrics gridding methods creates grids of information about the data on a node-by-node basis. The data metrics gridding methods are not, in general, weighted average interpolators of the Z values. For example, you can obtain information such as:

- the number of data points used to interpolate each grid node. If the number of data points used is fairly equal at each grid node, then the quality of the grid at each grid node can be interpreted.
- the standard deviation, variance, coefficient of variation, and median absolute deviation of the data at each grid node. These are measures of the variability in space of the grid, and are important information for statistical analysis.
- the distance to the nearest data point. For example, if the XY values of a data set are sampling locations, use the *Distance to Nearest* data metric to determine locations for new sampling locations. A contour map of the distance to the nearest data point, quantifies where higher future sampling density may be desired.

Data metrics use the local data set, including breaklines, for a specific grid node for the selected data metric. The local data set is defined by the search parameters. These search parameters are applied to each grid node to determine the local data set. In the following descriptions, when computing the value of a grid node (r,c), the local data set S(r,c) consists of data within the specified search parameters centered at the specific grid node only. The set of selected data at the current grid node (r,c), can be represented by S(r,c), where

$$S(r, c) = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)\}$$

and where n = number of data points in the local data set.

The Z(r,c) location refers to a specific node within the grid.

Planar Grids

Data metrics is used to provide information about your data. After information is obtained from data metrics, it is likely you will grid the data again using one of the other gridding methods. When using data metrics, you will usually want to use the same grid line geometry, search, breakline, and fault parameters as when you grid the data using another gridding method.

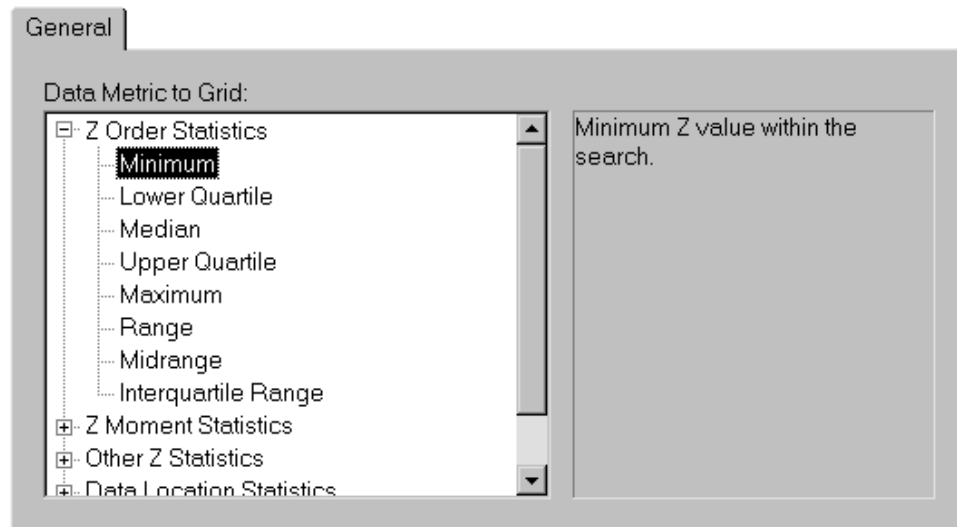
When using some data metrics, a horizontal planar or sloping planar grid is generated. This is usually a result of the selected search method. For example, consider using DEMOGRID.DAT and the *Count* data metric. The *Count* data metric determines the number of data points used in determining the grid node value. Since DEMOGRID.DAT contains 47 data points, *No search (use all of the data)* is the default search method. Using *No search (use all of the data)* means for each calculated grid node, all 47 points are used in determining the grid node value. The resulting data metric grid is horizontal planar because all grid nodes have a Z value of 47. The grid report shows both the Z minimum and the Z maximum as 47.

Other data metrics can yield similar results. For example, if the search radius is large enough to include all of the data using *Terrain Statistics*, the moving average is computed with such a large search radius that the resulting grid will be a planar surface at the data average. When interpreting data metrics results, keep the gridding parameters and the data metrics calculation approach (see the previous page) in mind.

Data Metrics Advanced Options

To set the data metrics advanced options:

1. Click on **Grid | Data**.
2. In the **Open** dialog, select a data file and then click the *Open* button.
3. In the **Grid Data** dialog, choose *Data Metrics* in the *Gridding Method* group.
4. Click the *Advanced Options* button to display the **Data Metrics Advanced Options** dialog.
5. Select a metric on the **General** page.
6. The **Search** page allows you to specify search rules. For more information on search, see the *Search* section beginning on page 103.
7. The **Breakline and Faults** page is used to add breaklines and faults to the gridding process. For more information on breaklines and faults, see the *Breaklines and Faults* section beginning on page 111.
8. Click OK in the **Data Metrics Advanced Options** dialog to return to the **Grid Data** dialog.
9. Click OK in the **Grid Data** dialog and a grid file is produced using the *Data Metrics* gridding method with the specified options.



*A description of the metric is listed in the right side of the **General** page when you click on each metric.*

Chapter 4 - Creating Grid Files

Data Metrics Descriptions

There are five groups of data metrics, *Z Order Statistics*, *Z Moment Statistics*, *Other Z Statistics*, *Data Location Statistics*, and *Terrain Statistics*.

Z Order Statistics

The data at a grid node within the search parameters are sorted for the Z order statistics.

$$\{z_{[1]}, z_{[2]}, \dots, z_{[n]}\}$$

where $z_{[1]} \leq z_{[2]} \leq \dots \leq z_{[n]}$. Square brackets indicate ordered Z values.

Data Metric	Definition	Equation
<i>Minimum</i>	each nodal value is the minimum Z value of data selected by the specified sector search centered at that node	$Z(r, c) = \min(z_1, z_2, \dots, z_n) = z_{[1]}$
<i>Lower Quartile</i>	each nodal value is the 25th percentile Z value of data selected by the specified sector search centered at that node	$Z(r, c) = z_{\left[\frac{n}{4}\right]}$
<i>Median</i>	each nodal value is the median Z value of data selected by the specified sector search centered at that node	$Z(r, c) = z_{\left[\frac{n}{2}\right]}$
<i>Upper Quartile</i>	each nodal value is the 75th percentile Z value of data selected by the specified sector search centered at that node	$Z(r, c) = z_{\left[\frac{3n}{4}\right]}$
<i>Maximum</i>	each nodal value is the maximum Z value of data selected by the specified sector search centered at that node	$Z(r, c) = \max(z_1, z_2, \dots, z_n) = z_{[n]}$
<i>Range</i>	each nodal value is the difference between the maximum Z value and the minimum Z value for the data selected by the specified sector search centered at that node	$Z(r, c) = z_{[n]} - z_{[1]}$
<i>Midrange</i>	each nodal value is the average of the maximum Z value and the minimum Z value, for the data selected by the specified sector search centered at that node	$Z(r, c) = \frac{z_{[n]} + z_{[1]}}{2}$
<i>Interquartile Range</i>	The <i>Interquartile Range</i> data metric generates a grid for which each nodal value is the difference between the 75th percentile Z value and the 25th percentile Z value, for the data selected by the specified sector search centered at that node. This data metric shows the spatial variation of variability of the data, but it focuses on the middle fifty percent of the data only. Thus, it is insensitive to variations in the tails of the local distributions.	$Z(r, c) = z_{\left[\frac{3n}{4}\right]} - z_{\left[\frac{n}{4}\right]}$

Z Moment Statistics

Data Metric	Definition	Equation
<i>Mean</i>	each nodal value is the arithmetic average of the data selected by the specified sector search centered at that node	$Z(r, c) = \frac{1}{n} \sum_{i=1}^n z_i$
<i>Standard Deviation</i>	each nodal value is the standard deviation of the data selected by the specified sector search centered at that node	$Z(r, c) = \sqrt{\frac{1}{n} \sum_{i=1}^n (z_i - \bar{z})^2}$ <p>where \bar{z} is the mean of the selected data</p>
<i>Variance</i>	each nodal value is the variance of the data selected by the specified sector search centered at that node	$Z(r, c) = \frac{1}{n} \sum_{i=1}^n (z_i - \bar{z})^2$ <p>where \bar{z} is the mean of the selected data</p>
<i>Coef. of Variation</i>	The <i>Coef. of Variation</i> data metric generates an output grid for which each nodal value is the local standard deviation divided by the local mean of the data selected by the specified sector search centered at that node. Note that this measure is useless for data whose local mean values are close to zero; this includes data that changes sign within the domain of interest.	$Z(r, c) = \frac{\text{standard deviation}(z)}{\text{mean}(z)}$

Chapter 4 - Creating Grid Files

Other Z Statistics

Data Metric	Definition	Equation
<i>Sum</i>	The <i>Sum</i> data metric generates an output grid for which each nodal value is the sum of the Z values of the data selected by the specified sector search centered at that node.	$Z(r, c) = \sum_{i=1}^n z_i$
<i>M.A.D.</i>	The Median Absolute Deviation (<i>M.A.D.</i>) data metric generates an output grid for which each nodal value is the median absolute deviation of the data selected by the specified sector search centered at that node.	$Z(r, c) = d_{\left[\frac{n}{2}\right]}$ <p>where $d_i = \left z_i - z_{\left[\frac{n}{2}\right]} \right$</p> <p>these d_i are sorted into ascending order and indexed as</p> $\{d_{[1]}, d_{[2]}, \dots, d_{[n]}\}$
<i>R.M.S.</i>	The Root Mean Square (<i>R.M.S.</i>) data metric generates an output grid for which each nodal value is the root mean square of the data selected by the specified sector search centered at that node.	$Z(r, c) = \sqrt{\frac{1}{n} \sum_{i=1}^n z_i^2}$

Data Location Statistics

The separation distances between the current grid node and each of the selected data are used in the computation of the data location statistics. In the following discussion, let the location of the current grid node be represented as (x_0, y_0) . The list of separation distances is defined as

$$R_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

The R_i are sorted in ascending order and indexed as

$$\{R_{[1]}, R_{[2]}, \dots, R_{[n]}\}$$

Data Metric	Definition	Equation
<i>Count</i>	The <i>Count</i> data metric generates an output grid for which each nodal value is the number of data selected by the specified sector search centered at that node. Under most circumstances, the best interpolation results occur when the <i>Count</i> is approximately homogeneous throughout the area of interest.	$Z(r, c) = n$
<i>Approximate Density</i>	The <i>Approximate Density</i> data metric generates an output grid for which each nodal value is the number of data selected by the specified sector search centered at that node, divided by πR_n^2 where R_n is the distance from the node to the farthest selected datum. The area over which the density is computed is bounded by the distance to the farthest selected datum. As such, when the data are relatively sparse, the computed density is slightly over-stated.	$Z(r, c) = \frac{n}{\pi R_n^2}$
<i>Distance to Nearest</i>	each nodal value is the distance to the nearest datum selected by the specified sector search centered at that node	$Z(r, c) = R_{[1]}$
<i>Distance to Farthest</i>	each nodal value is the distance to the farthest datum selected by the specified sector search centered at that node	$Z(r, c) = R_{[n]}$
<i>Median Distance</i>	each nodal value is the median separation distance between the node and all of the data selected by the specified sector search centered at that node	$Z(r, c) = R_{\left[\frac{n}{2}\right]}$
<i>Average Distance</i>	each nodal value is the average separation distance between the node and all of the data selected by the specified sector search centered at that node	$Z(r, c) = \frac{1}{n} \sum_{i=1}^n R_i$
<i>Offset Distance</i>	each nodal value is the distance between the node and the centroid of all of the data selected by the specified sector search centered at that node	$Z(r, c) = \sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$ <p>where the centroid coordinates</p> $(x_c, y_c) \text{ are } x_c = \frac{1}{n} \sum_{i=1}^n x_i \text{ and }$ $y_c = \frac{1}{n} \sum_{i=1}^n y_i$

Chapter 4 - Creating Grid Files

Terrain Statistics

Terrain statistics are all based upon a locally fitted planar surface. For each grid node, the specified sector search is performed. Then, using ordinary least squares, the following equation is fitted to the selected data:

$$Z_i = Ax_i + By_i + C + \varepsilon_i$$

where ε_i is an error term.

Least squares fitting is carried out using the data coordinates and it ignores faults. A sector search and the subsequent least squares fit are carried out for each grid node.

These data metrics are similar to **Grid | Calculus** terrain modeling, however, data metrics work on a local subset of the data rather than the grid.

Data Metric	Definition	Equation
<i>Terrain Slope</i>	each nodal value is the terrain slope of the least-squares-fit plane of the data selected by the specified sector search centered at that node	$Z(r,c) = \text{RadiansToDegrees}(\text{atan}(\text{hypot}(A,B)))$
<i>Terrain Aspect</i>	each nodal value is the terrain aspect, as an angle from zero to 360 degrees, of the least-squares-fit plane of the data selected by the specified sector search centered at that node	$Z(r,c) = 270.0 - \text{RadiansToDegrees}(\text{atan2}(A,B))$

Local Polynomial

The *Local Polynomial* gridding method assigns values to grid nodes by using a weighted least squares fit with data within the grid node's search ellipse.

For each grid node, the neighboring data are identified by the user-specified sector search. Using only these identified data, a local polynomial is fit using weighted least squares, and the grid node value is set equal to this value. Local polynomials can be order 1, 2, or 3.

The form of these polynomials are:

Order 1

$$F(X, Y) = a + bX + cY$$

Order 2

$$F(X, Y) = a + bX + cY + dXY + eX^2 + fY^2$$

Order 3

$$F(X, Y) = a + bX + cY + dXY + eX^2 + fY^2 + gX^2Y + hXY^2 + iX^3 + jY^3$$

The weighted least squares function weights data closer to the grid node higher and data further away lower. The weighting function depends on the search ellipse, the power, and the specific data geometry. The actual calculations for the weights are somewhat involved. Define T_{XX} , T_{XY} , T_{YX} , and T_{YY} by

$$T_{XX} = \frac{\cos(\phi)}{R_1}$$

$$T_{XY} = \frac{\sin(\phi)}{R_1}$$

$$T_{YX} = \frac{-\sin(\phi)}{R_2}$$

$$T_{YY} = \frac{\cos(\phi)}{R_2}$$

where

ϕ is the angle of the search ellipse

R_1 is search radius 1

R_2 is search radius 2

Chapter 4 - Creating Grid Files

Define A_{XX} , A_{XY} , and A_{YY} by

$$\begin{aligned}A_{XX} &= T_{XX}^2 + T_{YX}^2 \\A_{XY} &= 2(T_{XX}T_{XY} + T_{YX}T_{YY}) \\A_{YY} &= T_{YY}^2 + T_{XY}^2\end{aligned}$$

Note that these values (A_{XX} , A_{XY} , and A_{YY}) are a function of the search ellipse parameters only. They are the same for all data and for all grid nodes.

Next, consider a datum at location (X_i, Y_i) and a grid node at location (X_0, Y_0) . Let

$$\begin{aligned}dX &= X_i - X_0 \\dY &= Y_i - Y_0\end{aligned}$$

then

$$R_i = \sqrt{A_{XX}dX^2 + A_{XY}dXdY + A_{YY}dY^2}$$

and finally,

$$W_i = (1 - R_i)^p$$

where W_i is the weight for datum i and p is the specified power.

Let the collection of neighboring data be enumerated as

$$\{(x_i, y_i, z_i) \text{ for } i = 1, 2, \dots, N\}$$

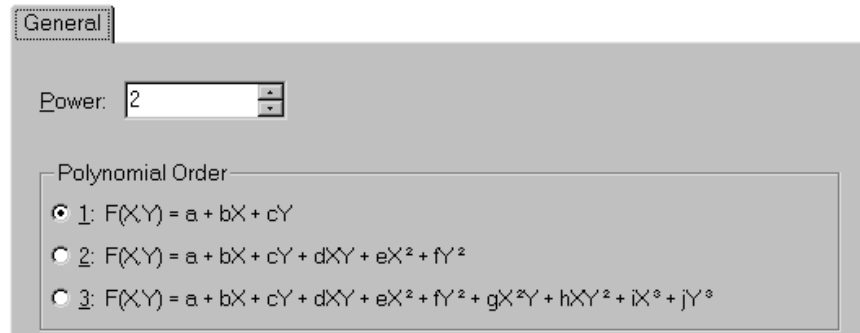
The local least squares parameters are computed by minimizing the weighted sum of the squared residuals:

$$\text{Minimize } \sum_{i=1}^N W_i [F(x_i, y_i) - z_i]^2$$

Local Polynomial Advanced Options

To set the local polynomial advanced options:

1. Click on **Grid | Data**.
2. In the **Open** dialog, select a data file and then click the *Open* button.
3. In the **Grid Data** dialog, choose *Local Polynomial* in the *Gridding Method* group.
4. Click the *Advanced Options* button to display the **Local Polynomial Advanced Options** dialog.
5. On the **General** page, set the *Power* to a number between 0 and 20, then select a *Polynomial Order*, 1, 2, or 3.
6. The **Search** page allows you to specify search rules. For more information on search, see the *Search* section beginning on page 103.
7. The **Breaklines** page is used to add breaklines to the gridding process. For more information on breaklines, see the *Breaklines and Faults* section beginning on page 111.
8. Click OK in the **Local Polynomial Advanced Options** dialog to return to the **Grid Data** dialog.
9. Click OK in the **Grid Data** dialog and a grid file is produced using the *Local Polynomial* gridding method with the specified options.



Choose a power and polynomial order in the **Local Polynomial Advanced Options** dialog.

Choosing Gridding Methods

Grid method parameters control the interpolation procedures. When you create a grid file, you can usually accept the default gridding method and produce an acceptable map. Different gridding methods provide different interpretations of your data because each method calculates grid node values using a different algorithm. If you are not satisfied with the map of your data, you might consider producing grid files using several different gridding methods and comparing the results.

Because **Surfer** maps are created from gridded data, the original data are not necessarily honored in the grid file. When you post the original data points on a contour map, some of the contour lines might be positioned "wrong" relative to the original data. This happens because the locations of the contour lines are determined solely by the interpolated grid node values and not directly by the original data. Some methods are better than others in preserving your data, and sometimes some experimentation (i.e. increasing grid density) is necessary before you can determine the best method for your data. See the *Gridding Methods* section of this chapter for more information on each method.

General Gridding Recommendations

The following list gives you a quick overview of each gridding method and some advantages and disadvantages in selecting one method over another.

- *Inverse Distance to a Power* is fast but has the tendency to generate "bull's-eye" patterns of concentric contours around the data points. Inverse distance to a power does not extrapolate Z values beyond the range of data.
- *Kriging* is one of the more flexible methods and is useful for gridding almost any type of data set. With most data sets, kriging with the default linear variogram is quite effective. In general, we would most often recommend this method. Kriging is the default gridding method because it generates a good map for most data sets. For larger data sets, kriging can be rather slow. Kriging can extrapolate grid values beyond your data's Z range.
- *Minimum Curvature* generates smooth surfaces and is fast for most data sets but it can create high magnitude artifacts in areas of no data. The *Internal Tension* and *Boundary Tension* allow you control over the amount of smoothing. Minimum curvature can extrapolate values beyond your data's Z range.
- *Modified Shepard's Method* is similar to inverse distance to a power but does not tend to generate "bull's eye" patterns, especially when a smoothing factor is used. Modified Shepard's method can extrapolate values beyond your data's Z range.
- *Natural Neighbor* generates good contours from data sets containing dense data in some areas and sparse data in other areas. It does not generate data in areas without data. Natural neighbor does not extrapolate Z grid values beyond the range of data.

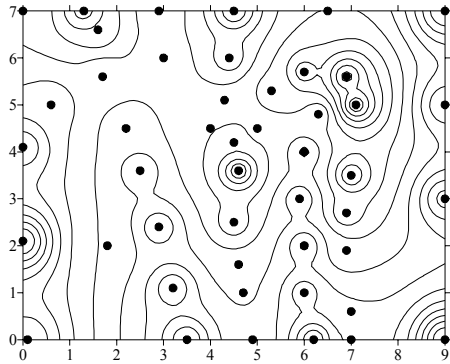
- *Nearest Neighbor* is useful for converting regularly spaced (or almost regularly spaced) XYZ data files to grid files. When your observations lie on a nearly complete grid with few missing holes, this method is useful for filling in the holes, or creating a grid file with the blanking value assigned to those locations where no data are present. *Nearest Neighbor* does not extrapolate Z grid values beyond the range of data.
- *Polynomial Regression* processes the data so that underlying large-scale trends and patterns are shown. This is used for trend surface analysis. Polynomial regression is very fast for any amount of data, but local details in the data are lost in the generated grid. This method can extrapolate grid values beyond your data's Z range.
- *Radial Basis Function* is quite flexible. It compares to kriging since it generates the best overall interpretations of most data sets. This method produces a result quite similar to kriging.
- *Triangulation with Linear Interpolation* is fast. When you use small data sets, triangulation with linear interpolation generates distinct triangular faces between data points. Triangulation with linear interpolation does not extrapolate Z values beyond the range of data.
- *Moving Average* is most applicable to large and very large data sets (e.g. > 1,000 observations). *Moving Average* extracts intermediate-scale trends and variations from large, noisy data sets, and it is fast even for very large data sets. This gridding method is a reasonable alternative to *Nearest Neighbor* for generating grids from large, regularly spaced data sets.
- *Data Metrics* is used to create grids of information about the data.
- *Local Polynomial* is most applicable to data sets that are locally smooth (i.e. relatively smooth surfaces within the search neighborhoods). The computational speed of the method is not significantly affected by the size of the data set.

The following figure shows a comparison of all the gridding methods applied to the file DEMOGRID.DAT. This data file contains 47 data points irregularly distributed over the extent of the map. The defaults were accepted for all the gridding methods.

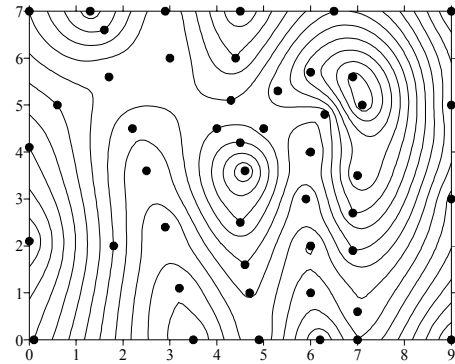
Chapter 4 - Creating Grid Files

It is recommended that you try each of the different methods, accepting the defaults, in much the same fashion as you have seen here. This gives you a way to determine the best gridding method to use with a data set.

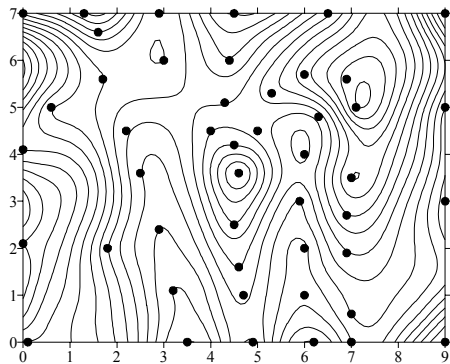
- With DEMOGRID.DAT, *Kriging*, *Minimum Curvature*, *Natural Neighbor*, and *Radial Basis Function* all produced acceptable contour maps with smooth appearance.
- *Inverse Distance to a Power* and *Modified Shepard's Method* both tended to generate "bull's eye" patterns.
- With *Triangulation with Linear Interpolation*, there are too few data points to generate an acceptable map, and this explains the triangular facets apparent on the contour map.
- *Nearest Neighbor* shows as a "blocky" map because the data set is not regularly spaced and is therefore a poor candidate for this method.
- *Polynomial Regression* shows the trend of the surface, represented as a tilted plane.
- Due to the small number of data in DEMOGRID.DAT, the *Moving Average* method is not applicable. The results of using this method with an inadequate data set are shown as discontinuities are created as data are captured and discarded by the moving search neighborhoods.
- *Data Metrics* can show many different types of information about the data and about the gridding process, depending on which metric is selected. In this case, the median distance metric was selected, and it shows the median distance between each grid node and the original 47 data points.
- *Local Polynomial* models smooth local variation in the data set.



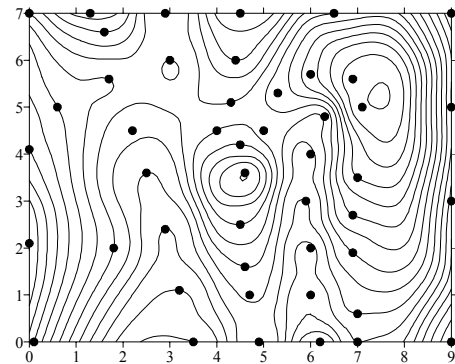
Inverse Distance to a Power



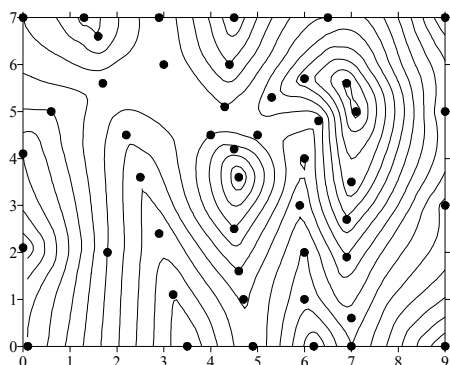
Kriging



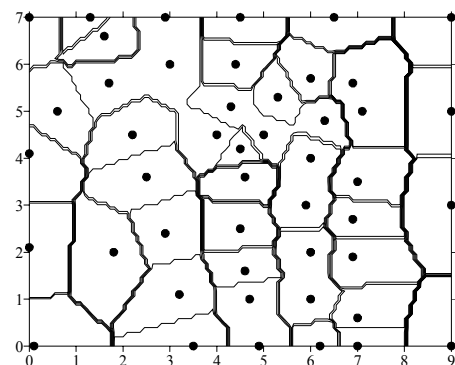
Minimum Curvature



Modified Shepard's Method



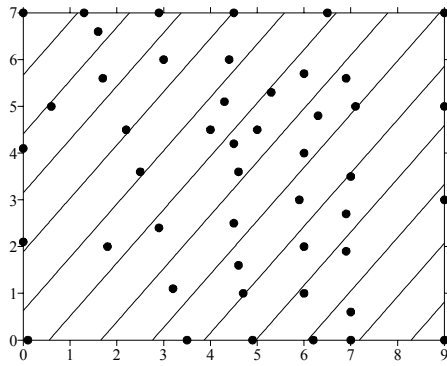
Natural Neighbor



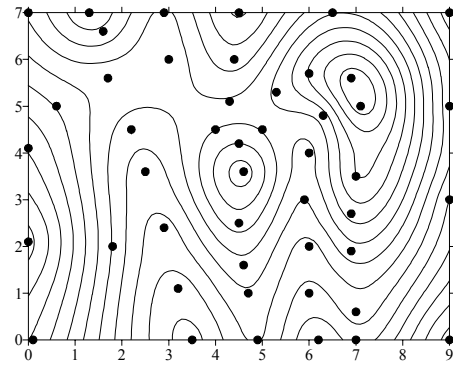
Nearest Neighbor

This is a comparison of the different gridding methods. For these examples, the same file, DEMOGRID.DAT, was used. All the defaults for the various methods were accepted. This data set contains 47 data points, irregularly spaced over the extent of the map. The data point locations are indicated with dots on the maps.

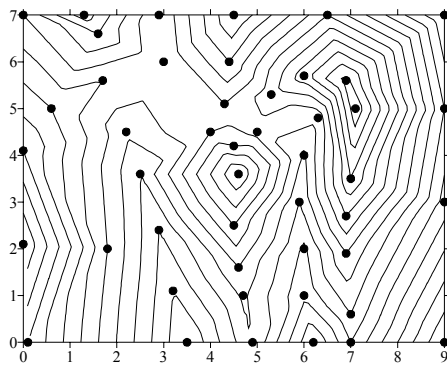
Chapter 4 - Creating Grid Files



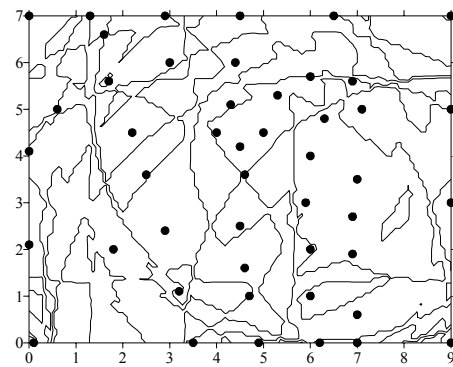
Polynomial Regression



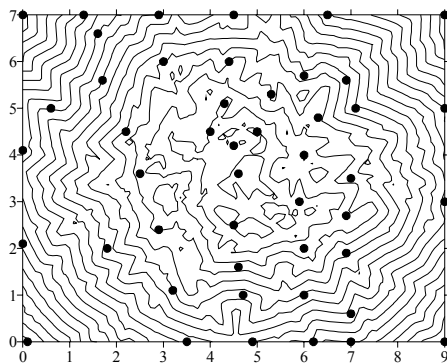
Radial Basis Function



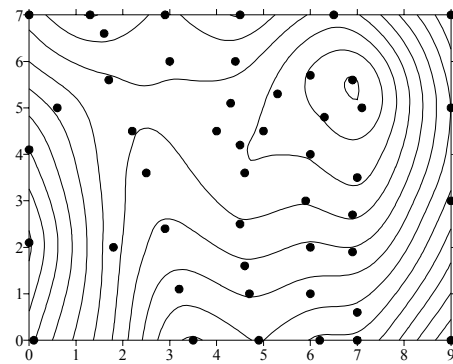
Triangulation with Linear Interpolation



Moving Average



Data Metrics



Local Polynomial

This is a comparison of the different gridding methods. For these examples, the same file, DEMOGRID.DAT, was used. All the defaults for the various methods were accepted. This data set contains 47 data points, irregularly spaced over the extent of the map. The data point locations are indicated with dots on the maps.

Choosing Methods Based on the Number of XYZ Data Points

The size of your input data set should be considered when selecting a gridding method. For example, some gridding methods interpret small data sets more effectively than others do. **Surfer** needs a bare minimum of three non-collinear XYZ points to perform the gridding process.

- Ten or fewer points are not enough to define more than a general trend in your data. *Triangulation with Linear Interpolation* and *Moving Average* are not effective with few points. As with most data sets, *Kriging* and *Radial Basis Function* methods will produce the best representation of your data in this situation. If you want only to define the trend of the data, you can use *Polynomial Regression*. With 10 or fewer points, gridding is extremely fast, so you might want to try the different methods to determine the most effective method for your data.
- With small data sets (<250 observations), *Kriging* with the default linear variogram, or *Radial Basis Function* with the *multiquadric* function produce good representations of most data sets.
- With moderate-sized data sets (from 250 to 1000 observations), *Triangulation with Linear Interpolation* is fast and creates a good representation of your data. Although *Kriging* or *Radial Basis Function* generate the grids more slowly, they also produce good data representations.
- For large data sets (>1000 observations), both *Minimum Curvature* and *Triangulation with Linear Interpolation* are quite fast, and both produce good representations. As with most other data sets, *Kriging* or *Radial Basis Function* probably produce the best maps but are quite a bit slower.
- Using *Kriging* or *Radial Basis Function* with large data sets does not result in significantly different gridding times. For example, if your data file contains 3,000 or 30,000 data points, the gridding time is not significantly different. Either data set might take a considerable amount of time to grid, but they take approximately the same amount of time.

Exact and Smoothing Interpolators

Gridding methods included with **Surfer** are divided into three general categories: exact interpolators, smoothing interpolators, and data metrics. Some exact interpolators can incorporate a smoothing factor that causes them to become smoothing interpolators.

Exact interpolators honor data points exactly when the point coincides with the grid node being interpolated. In other words, a coincident point carries a weight of essentially 1.0 and all other data points carry a weight of essentially zero. Even when using exact interpolators, it is possible that the grid file does not honor specific data points if the data points do not exactly coincide with the grid nodes. Refer to the *Weighted Averaging* section of this chapter for more information on weights assigned during interpolation.

To increase the likelihood that your data are honored, you can increase the number of grid lines in the X and Y direction. This increases the chance that grid nodes coincide with data points, thereby increasing the chance that the data values are applied directly to the grid file.

The following methods are exact interpolators:

- *Inverse Distance to a Power* when you do not specify a smoothing factor
- *Kriging* when you do not specify a nugget effect
- *Nearest Neighbor* under all circumstances
- *Radial Basis Function* when you do not specify an R^2 value
- *Modified Shepard's Method* when you do not specify a smoothing factor
- *Triangulation with Linear Interpolation*
- *Natural Neighbor*

Smoothing interpolators or smoothing factors can be employed during gridding when you do not have strict confidence in the repeatability of your data measurements. This type of interpolation reduces the effects of small-scale variability between neighboring data points. Smoothing interpolators do not assign weights of 1.0 to any single point, even when a point is exactly coincident with the grid node. When smoothing is used, weighting factors are assigned so the map is smoother. In the extreme case, all data points are given equal weight and the surface becomes a horizontal plane at the average for all data in the data file.

The following methods are smoothing interpolators:

- *Inverse Distance to a Power* when you specify a smoothing factor
- *Kriging* when you specify an error nugget effect
- *Polynomial Regression*
- *Radial Basis Function* when you specify an R^2 value
- *Modified Shepard's Method* when you specify a smoothing factor
- *Local Polynomial*
- *Moving Average*

Weighted Averaging

The gridding methods in **Surfer** use weighted average interpolation algorithms, with the exception of data metrics. This means that, with all other factors being equal, the closer a point is to a grid node, the more weight it carries in determining the Z value at that grid node. The difference between gridding methods is how the weighting factors are computed and applied to data points during grid node interpolation.

To understand how weighted averages are applied, consider the equation shown here. Given N data values:

$$\{Z_1, Z_2, \dots, Z_N\}$$

the interpolated value at any grid node (for example, G_j) can be computed as the weighted average of the data values:

$$G_j = \sum_{i=1}^N w_{ij} Z_i$$

where

G_j is the interpolated grid node value at node j

N is the number of points used to interpolate at each node

Z_i is the Z value at the i^{th} point

w_{ij} is the weight associated with the i^{th} data value when computing G_j

Producing a Grid File from a Regular Array of XYZ Data

When your XYZ data are collected or generated on regular intervals it is possible to produce a grid file that uses your values directly and does not interpolate the values for the grid nodes. You can use the **Grid | Data** command to create a grid file in the correct format.

When you have a complete array of XYZ data (or a nearly complete set of data with only a few "holes"), you can use the nearest neighbor gridding method to convert your data directly to a grid file. The nearest neighbor method does not interpolate data but merely picks up the closest point and assigns that value to the grid.

To use the **Data** command to produce a grid file from regularly spaced data:


1. Create an XYZ data [.DAT] file from your data.
2. Choose the **Grid | Data** command.
3. The **Open** dialog is displayed. Specify the name of the XYZ data file, click OK, and the **Grid Data** dialog is displayed.
4. In the *Grid Line Geometry* group, set the *Spacing* values to match the spacing of your data in the X and Y directions. This assures that the grid nodes coincide with your data. For example, if your data are separated by 100 meters over the extent of your map, enter 100 for the *X Direction* and *Y Direction Spacing* boxes.
5. Select *Nearest Neighbor* as the *Gridding Method*. During gridding, the nearest Z value from the data file is assigned to the grid node.
6. Click OK in the **Grid Data** dialog and the grid file is created.

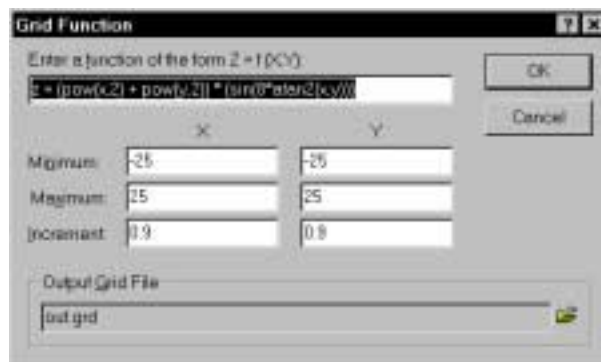
You can also produce grid files directly from an evenly spaced array of Z values. When your Z values are organized correctly in an ASCII file, you can add some header information identifying the data as a grid file, specify the limits of the data, and then save the file. The ASCII grid file format is given in *Appendix C, File Formats*.

Gridding a Function

The **Grid | Function** command allows you to create a grid file from a user-defined two variable equation of the form $Z = f(X, Y)$. The density of the generated grid is a function of the *Minimum*, *Maximum*, and *Increment* values for both X and Y. The **Function** command can use any of the mathematical functions listed in *Appendix A, Mathematical Functions*. Grids created as functions are plotted in the same way as grids created using the **Data** command.

To grid a function:

1. Click the **Grid | Function**. The **Grid Function** dialog is displayed.
2. Enter the function of interest into the box titled: *Enter a function of the form $Z = f(X, Y)$* . The function calculation is repeated one time for each Z value to be written to the output grid file. The number of grid nodes in the output grid and, therefore, the number of calculations to be performed, is based on the *Minimum*, *Maximum* and *Increment* values specified.
3. In the *Minimum* X and Y boxes, specify the beginning values used in the specified function. These values are entered into the defined function and the first grid node is calculated based on these values. These values also specify the lower limits for X and Y in the output grid file.
4. In the *Maximum* X and Y boxes, specify upper values to be applied to the function and the upper X and Y limits for the grid.
5. In the *Increment* boxes, specify the amount to step in the X and Y directions for each grid line. This is similar to the grid spacing setting in the **Grid Data** dialog.
6. In the *Output Grid File* group, click the  button and specify the path and file name for the grid file to be created.
7. Click OK and a grid file is created based on the function and parameters you specify.



You can set the mathematical function, the minimum and maximum X and Y values, and the function increment values in the **Grid Function** dialog.

Grid Files

In addition to the grid files that **Surfer** creates, you can also use Digital Elevation Models (DEM) such as USGS DEM [.DEM], GTopo30 [.HDR], SDTS [.DDF], and DTED [.DT*] grid files. DEM files containing arc-second XY coordinates are automatically converted to latitude/longitude decimal degrees.

GRD

To preserve faulting information and to use double precision values in **Surfer**, be sure to save the grid in the *Surfer 7* format. If you need to use a grid in **Surfer 5** or **Surfer 6**, save the grid as *GS ASCII* [.GRD] or *GS Binary* [.GRD]. Keep in mind that these two formats do not retain fault information or double precision values. See *Appendix C, File Formats* for detailed information on **Surfer** grid files.

USGS DEM

The United States Geological Survey (USGS) produced two DEM file types: the 7.5 minute DEM data (1:24,000 scale) and the 1:250,000-scale DEM data (1:250,000 scale). These files have been replaced by the newer SDTS format, although you can still use these files in **Surfer**.

GTopo30

USGS GTopo30 files are extremely large. **Surfer** can read the files, though approximately 220 MB of RAM is needed for this operation due to the grid size. You can use **Grid | Extract** to make the file smaller. You must have the associated [.DEM], [.HDR], and [.STX] files to use GTopo30 files in **Surfer**. These files are necessary for computing the X, Y, and Z limits. Some GTopo30 files contain a "no data" value of -9999 in the file. The no data value is set to zero elevation in **Surfer**.

SDTS

There are two separate types of USGS SDTS files. The Topological Vector Profile SDTS files contain boundary line information and can be used as a base map (use with **File | Import** or **Map | Base Map**). The Raster Profile SDTS files contain gridded elevation information and can be used to create a grid-based map. An uncompressed SDTS DEM contains several files with the [.DDF] extension. You can choose any one of these files in the **Open Grid** dialog when creating a grid-based map. All of the [.DDF] files are necessary to produce a map (i.e. you cannot copy just one [.DDF] file and create a map from it).

DTED

Digital Terrain Elevation Data (DTED) are grids originally produced by the United States National Imaging and Mapping Agency (NIMA). The files contain latitude/longitude coordinates with the Z units in meters.

There are multiple "levels" of DTED files. Level 0 files [.DT0] are 1-degree square blocks with 30 arc-second spacing, similar to GTOPO 30 files. Level 1 files [.DT1] use three arc-second spacing. Level 2 files [.DT2] use one arc-second spacing.

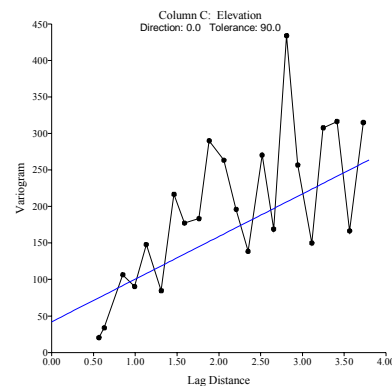
Chapter 5

Variograms

Introduction to Variograms

Surfer includes an extensive variogram modeling subsystem. This capability was added to **Surfer** as an integrated data analysis tool. The primary purpose of the variogram modeling subsystem is to assist users in selecting an appropriate variogram model when gridding with the kriging algorithm. The variogram modeling subsystem may also be used to quantitatively assess the spatial continuity of data even when the kriging algorithm is not applied.

Surfer's variogram modeling feature is intended for experienced variogram users. The novice variogram user may find the following four authors helpful: Cressie (1991), Isaaks and Srivastava (1989), Kitanidis (1997), and Pannatier (1996). Please refer to the *Suggested Reading* section of this chapter for full references for each of the previous books. If you do not understand variograms or if you are unsure about which model to apply, use **Surfer's** default linear variogram with the kriging algorithm.



This is a variogram using the default parameters.

Variogram modeling is not an easy or straightforward task. The development of an appropriate variogram model for a data set requires the understanding and application of advanced statistical concepts and tools: this is the *science* of variogram modeling. In addition, the development of an appropriate variogram model for a data set requires knowledge of the tricks, traps, pitfalls, and approximations inherent in fitting a theoretical model to real world data: this is the *art* of variogram modeling. Skill with the science and the art are both necessary for success.

The development of an appropriate variogram model requires numerous correct decisions. These decisions can only be properly addressed with an intimate knowledge of the data at hand, and a competent understanding of the data genesis (i.e. the underlying processes from which the data are drawn). The cardinal rule when modeling variograms is *know your data*.

Variogram Components

Kriging and Variograms

The kriging algorithm incorporates four essential details:

1. When computing the interpolation weights, the algorithm considers the spacing between the point to be interpolated and the data locations. The algorithm considers the inter-data spacing as well. This allows for declustering.
2. When computing the interpolation weights, the algorithm considers the inherent *length scale* of the data. For example, the topography in Kansas varies much more slowly in space than does the topography in central Colorado. Consider two observed elevations separated by five miles. In Kansas, it would be reasonable to assume a linear variation between these two observations, while in the Colorado Rockies such an assumed linear variation would be unrealistic. The algorithm adjusts the interpolation weights accordingly.
3. When computing the interpolation weights, the algorithm considers the inherent trustworthiness of the data. If the data measurements are exceedingly precise and accurate, the interpolated surface goes through each and every observed value. If the data measurements are suspect, the interpolated surface may not go through an observed value, especially if a particular value is in stark disagreement with neighboring observed values. This is an issue of *data repeatability*.
4. Natural phenomena are created by physical processes. Often these physical processes have preferred orientations. For example, at the mouth of a river the coarse material settles out fastest, while the finer material takes longer to settle. Thus, the closer one is to the shoreline the coarser the sediments, while the further from the shoreline the finer the sediments. When computing the interpolation weights, the algorithm incorporates this natural *anisotropy*. When interpolating at a point, an observation 100 meters away but in a direction parallel to the shoreline is more likely to be similar to the value at the interpolation point than is an equidistant observation in a direction perpendicular to the shoreline.

Items two, three, and four incorporate something about the underlying process from which the observations were taken. The *length scale*, *data repeatability*, and *anisotropy* are not a function of the data locations. These enter into the kriging algorithm via the variogram. The *length scale* is given by the variogram **range** (or **slope**), the *data repeatability* is specified by the **nugget effect**, and the *anisotropy* is given by the **anisotropy**.

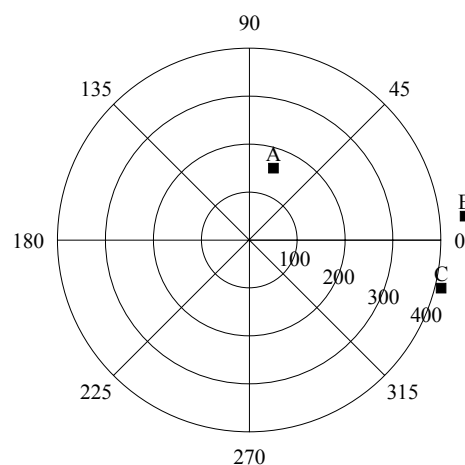
The Variogram Grid

Users familiar with GeoEAS or VarioWin® should be familiar with pair comparison files [.PCF]. **Surfer** uses a variogram grid as a fundamental internal data representation, in lieu of a pair comparison file. The pair comparison file can be extremely large for moderately sized data sets. For example, 5000 observations create $N(N-1)/2$ pairs (12,497,500). Each pair requires 16 bytes of information for a pair comparison file, so a 5000-observation pair comparison file would take approximately 191 megabytes of memory to merely hold the pair comparison information. The time to read and search through this large file makes this approach impractical for many **Surfer** users.

Computational speed and storage are gained by using the variogram grid approach. Once the variogram grid is built, any experimental variogram can be computed instantaneously. This is independent of the number of observations. However, the ability to carry out on-the-fly editing of variograms on a pair-by-pair basis is lost by using the variogram grid approach in **Surfer**.

Unlike the grids used elsewhere in **Surfer**, which are rectangular grids, variogram grids are polar grids. Polar grids cannot be viewed in **Surfer**, and are only used within the context of variogram computation. The first coordinate in a variogram grid is associated with the polar angle, and the second coordinate is associated with the radial distance out from the origin.

Consider the following variogram grid:



This is a polar variogram grid for three pairs (A, B, and C) of observation points, showing separation angles and separation distances.

Chapter 5 - Variograms

There are eight angular divisions: $\{0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ\}$ and four radial divisions: $\{100, 200, 300, 400\}$. Thus, there are 32 individual cells in this variogram grid. Users familiar with VarioWin[®] will notice similarities between **Surfer's** variogram grid and the "variogram surface" in VarioWin[®] 2.2. In **Surfer**, only the upper half of the grid is used. See the *General* section of this chapter for a more detailed explanation.

Consider the following three observation locations: $\{(50,50), (100, 200), \text{ and } (500,100)\}$. There are three observations, so there are $3*(3-1)/2 = 3$ pairs. The pairs are:

A (50,50), (100,200)

B (50,50), (500,100)

C (100,200), (500,100)

Each pair is placed in a particular cell of the variogram grid based upon the separation distance and separation angle between the two observation locations.

The separation distance is:

$$h = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

The separation angle is:

$$\theta = \arctan \left[\frac{Y_2 - Y_1}{X_2 - X_1} \right]$$

Using the previous equations, the separation angle for the first pair of observations $\{(50,50), (100,200)\}$ is 71.57 degrees and the separation distance is 158.11. This pair is placed in the cell bounded by the 100 circle on the inside, the 200 circle on the outside, the 45° line in the clockwise direction, and the 90° line in the counterclockwise direction. The location of this pair in the variogram grid is shown on the previous page as point A.

Pair	Separation Angle	Separation Distance
A	71.57	158.11
B	6.34	452.77
C	-14.04	412.31

The separation angle and separation distance for each pair are calculated using the previous equations.

Since the separation distance of pairs B and C are greater than the radius of the largest circle (400), these pairs fall outside of the variogram grid. Pairs B and C are not included in the variogram grid and therefore, not included in the variogram. Using the previous equations, every pair is placed into one of the variogram grid cells or it is discarded if the separation distance is too large.

For a large data set, there could be millions of pairs (or more) and the associated pair comparison file would be very large. On the other hand, with the variogram grid in the example above there are only 32 grid cells regardless of the number of pairs contained in a particular grid cell. Herein lies the computational saving of the variogram grid approach. It is not necessary that every pair is stored in a variogram grid cell; each variogram grid cell stores only a small set of summary statistics that represent all of the pairs contained within that cell.

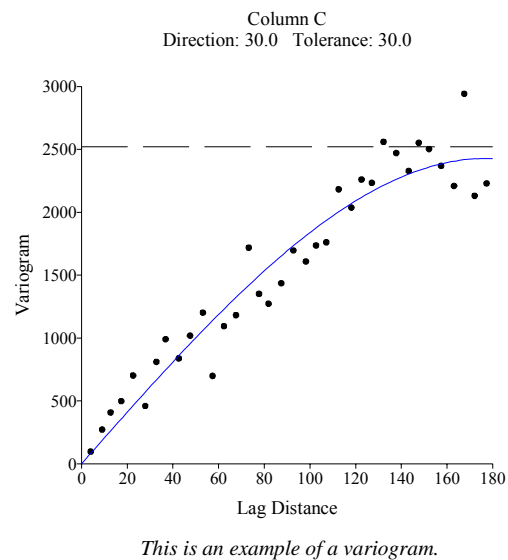
The Variogram

The variogram is a measure of how quickly things change on the average. The underlying principle is that, on the average, two observations closer together are more similar than two observations farther apart. Because the underlying processes of the data often have preferred orientations, values may change more quickly in one direction than another. As such, the variogram is a function of direction.

The variogram is a three dimensional function. There are two independent variables (the direction θ , the separation distance h) and one dependent variable (the variogram value $\gamma(\theta, h)$). When the variogram is specified for kriging we give the sill, range, and nugget, but we also specify the anisotropy information. The variogram grid is the way this information is organized inside the program.

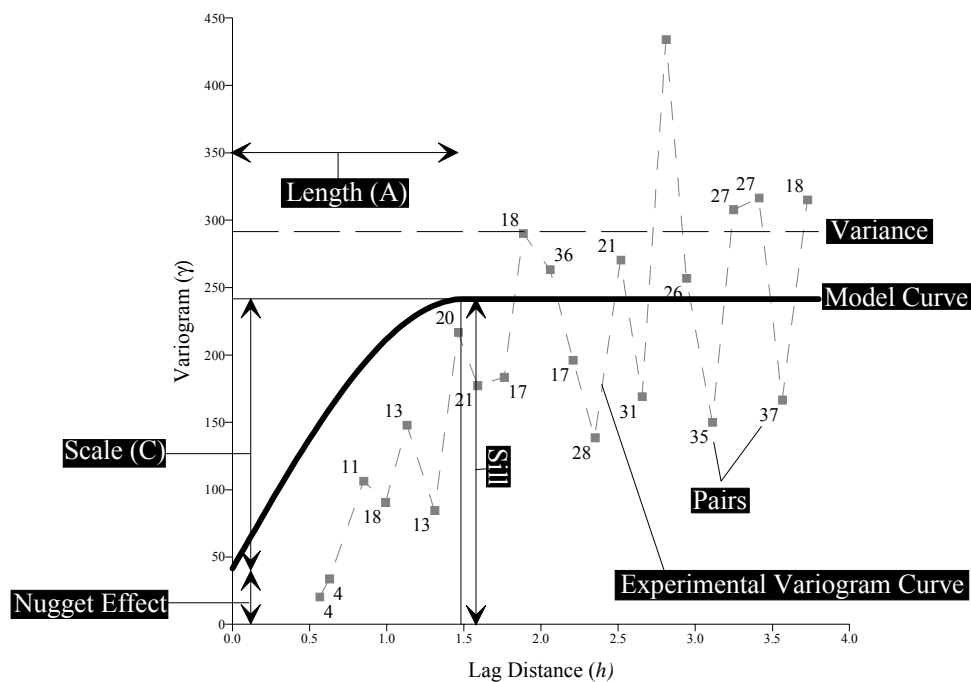
The variogram (XY plot) is a radial slice (like a piece of pie) from the variogram grid, which can be thought of as a "funnel shaped" surface. This is necessary because it is difficult to draw the three-dimensional surface, let alone try to fit a three dimensional function (model) to it. By taking slices, it is possible to draw and work with the directional experimental variogram in a familiar form - an XY plot.

Remember that a particular directional experimental variogram is associated with a direction. The ultimate variogram model must be applicable to all directions. When fitting the model, the user starts with numerous slices, but must ultimately mentally integrate the slices into a final 3D model.



The Variogram Model

The variogram model mathematically specifies the spatial variability of the data set and the resulting grid file. The interpolation weights, which are applied to data points during the grid node calculations, are direct functions of the variogram model.



This is a variogram model demonstrating the variogram parameters.

NUGGET EFFECT: quantifies the sampling and assaying errors and the short scale variability (i.e. spatial variation occurring at distance closer than the sample spacing).

SCALE (C): is the vertical scale for the structured component of the variogram. Each component of a variogram model has its own scale.

SILL: is the total vertical scale of the variogram (*Nugget Effect + Sum of all component Scales*). *Linear, Logarithmic, and Power* variogram models do not have a sill.

Chapter 5 - Variograms

LENGTH: is the horizontal range of the variogram. (Some variogram models do not have a length parameter; e.g., the linear model has a slope instead.)

VARIANCE: is the mean squared deviation of each value from the mean value. Variance is indicated by the dashed horizontal line in the diagram shown above.

$$\text{var} = \frac{\sum_{i=1}^N z_i^2}{N} - \left(\frac{\sum_{i=1}^N z_i}{N} \right)^2 \quad \text{where: } N \text{ is the number of data}$$

PAIRS: represents the average variogram value for the group of pairs separated by a specified distance (lag width). The number adjacent to the square symbols indicates the number of pairs within each lag distance.

MODEL CURVE: shows the shape of the variogram model.

EXPERIMENTAL CURVE: displays the groups of variogram pairs on a plot of separation distance versus the estimated variogram.

For more information on the nugget effect, scale, and length settings, see the *Model* section of this chapter.

Creating a Variogram

When you create a new variogram, the **New Variogram** dialog is displayed after the data file is selected. The **New Variogram** dialog contains options for setting up the variogram grid used for the variogram. Once the variogram is created through the **New Variogram** dialog, the variogram properties, such as the variogram model used, are set through the variogram properties dialog. Once the variogram grid is created, it is used for the duration of the variogram modeling effort. See *The Variogram Grid* section for more information on variogram grids.

The following directions contain the general steps to create a variogram. Decisions about the various options should be made based on knowledge of the data and based on knowledge of variograms. As such, specific modeling recommendations cannot be made.

1. Choose **Grid | Variogram | New Variogram**.
2. Select a data file from the **Open** dialog and click the *Open* button.
3. Set the data columns to use on the **Data** page in the **New Variogram** dialog, and select the variogram grid settings on the **General** page.
4. Once the variogram has been created, using one of the methods described in the **Editing Variogram Properties** section. The variogram properties dialog allows you to select a variogram model other than the default linear model, and it allows you to set several other options discussed later in this chapter.

Editing Variogram Properties

To change the features of the variogram, such as the variogram model, open the variogram properties dialog by:

- double-clicking on the variogram in the plot window,
- right-clicking the variogram and choosing **Properties** in the context menu,
- right-clicking on the variogram in the **Object Manager** and choosing **Properties** in the context menu,
- or by clicking **Edit | Properties** when the variogram is selected.

New Variogram Properties

The **New Variogram** dialog contains two pages: **Data** and **General**.

Data

You can select data for a new variogram on the **Data** page in the **New Variogram** dialog.

Data Columns

You can specify the columns for the X data, the Y data, and the Z data in the *Data Columns* group. **Surfer** defaults to X: Column A, Y: Column B, and Z: Column C. Your data can be in any three columns, however. Click the down arrow on each box and select the appropriate column for each variable.

Statistic	X	Y	Z
Active	24	24	24
Original	24	24	24
Excluded	0	0	0
Deleted Dup.	0	0	0

*The **Data** page in the **New Variogram** dialog is used to set the data columns, filter duplicate data, and to view data statistics.*

Duplicate Data

You can choose the type of duplicate data to use in the variogram gridding process from the *To Keep* list. The *X Tolerance* and *Y Tolerance* boxes set the distance, in data units, that define whether or not the data are duplicates. If the data do not share identical coordinates, but you would like close-by points to be considered duplicates, use this setting. Refer to the *Filter Data* section in *Chapter 4, Creating Grid Files* for more information on duplicate data.

Data Exclusion Filter

To set up rules for excluding data when gridding, use the *Data Exclusion Filter*. Refer to the *Filter Data* section in *Chapter 4, Creating Grid Files* for more information on data exclusion.

Statistics

The bottom of the dialog displays statistics about the data. If you change any of the parameters in the dialog, click the *Update Statistics* button to display statistics based on the changed information.

General

The **General** page is used to define the variogram grid.

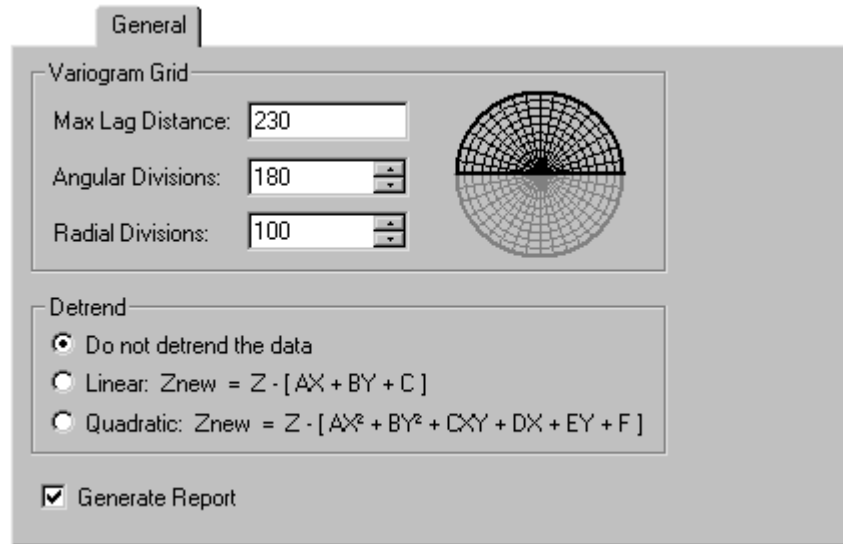
Variogram Grid

The *Variogram Grid* group contains the maximum lag distance, angular divisions, and the radial divisions settings for the variogram grid, as pictured on the right side of the group.

Max Lag Distance specifies the largest separation distance contained in the variogram grid. Any pairs separated by more than the maximum lag distance are not included in the variogram grid. The default value of the *Max Lag Distance* is one-third the diagonal extent of the data rounded down at the second significant figure (i.e. 1.2345 becomes 1.2).

The *Angular Divisions* specify the number of angular divisions: i.e. the number of spaces between "spokes" in the variogram grid. 0° is located on the positive X axis, not the positive Y axis (azimuth). The angular divisions only go from 0° to 180°, and not all the way back around to 360°. If a vector were drawn from the first point in the pair to the second point in the pair, the arrow symbol would point to one direction in the polar grid. In **Surfer**, the first point is the point with the smaller Y value and the second point is the point with the larger Y value. As such, the lower half of the variogram grid is empty and unnecessary. The experimental variogram for any direction between 180° and 360° can be computed from the upper half of the grid using the inherent symmetry of the variogram. The default value of 180 makes each variogram grid cell span 1° of arc.

The *Radial Divisions* specify the number of concentric circles in the variogram grid. The default value is 100. Keep in mind that increasing the number of *Angular Divisions* and *Radial Divisions* increases the amount of memory required to store the grid. See *The Variogram Grid* section for more information on memory requirements.



General

Variogram Grid

Max Lag Distance: 230

Angular Divisions: 180

Radial Divisions: 100

Detrend

☒ Do not detrend the data

☐ Linear: $Z_{\text{new}} = Z - [AX + BY + C]$

☐ Quadratic: $Z_{\text{new}} = Z - [AX^2 + BY^2 + CXY + DX + EY + F]$

☒ Generate Report

*You can set the variogram grid properties on the **General** page in the **New Variogram** dialog.*

The picture to the right of the edit controls depicts the specified variogram grid. However, to make the picture legible, the number of concentric circles drawn is equal to the number of radial divisions divided by ten. The number of radial spokes drawn is equal to the number of angular divisions divided by ten.

Detrend

The *Detrend* group contains three options for detrending the data before creating a variogram. The *Detrend* options carry out a simple polynomial least squares regression of the data and computes the variogram grid for the resulting residuals only. See Section 6.12 in Kitanidis (1997) for a more detailed discussion of detrending data.

- *Do not detrend the data* should be used most of the time - it should be used all of the time by novices.
- *Linear* detrending should be used when the kriging algorithm is applied with a linear drift.
- *Quadratic* detrending should be used when the kriging algorithm is applied with a quadratic drift.

Reports

Click in the *Generate Report* check box to create a report of the data, including statistics. For more information about the contents of the report, see the *Reports* topic in online help.

Variogram Properties

Once the variogram grid has been created with **Grid | Variogram | New Variogram**, the properties of the variogram can be changed through the variogram properties dialog. The variogram properties dialog contains four pages: **Experimental**, **Model**, **Statistics**, and **Plot**.

Experimental

The **Experimental** page in the variogram properties dialog contains parameters for the experimental variogram such as the lag direction, estimator type, maximum lag distance, number of lags, lag width, and vertical scale.

Lag Direction

The *Lag Direction* group controls the direction from which the variogram is viewed. The *Direction* specifies the focal direction when computing the experimental variogram. In keeping with the standard notation used in geostatistics, this direction is given using a mathematical convention: 0° is along the positive X axis, and 90° is along the positive Y axis. The direction is not given as an azimuth.

Tolerance

The direction *Tolerance* specifies the size of the angular window for the experimental variogram. The angular window is:

$$\text{Direction} - \text{Tolerance} < \text{Angle} < \text{Direction} + \text{Tolerance}$$

thus, the width of the entire angular window is twice the *Tolerance*. For example, the default *Tolerance* of 90° captures all directions (omni-directional variogram). The diagrammatic representation of the current angular window appears in the *Lag Direction* group.

Step Amount and Step Direction

When building a complete model of a variogram, it is necessary to review the variogram and the model in many directions. **Surfer** allows you to step through as many directions as desired in an animated fashion. Since **Surfer** uses a variogram grid approach, it can compute and replot directional variograms almost instantaneously. The *Step Amount* specifies the increment in the lag *Direction* each time, the *Step CW* or *Step CCW* buttons are clicked. When the *Step CW* button is clicked, the lag *Direction* is decremented by the *Step Amount* and the variogram plot is automatically updated. When the *Step CCW* button is clicked, the lag *Direction* is incremented by the *Step Amount* and the variogram plot is automatically updated. You can drag the variogram properties dialog away from the variogram to see the changes each time the *Step CW* or *Step CCW* buttons are clicked.

Estimator Type

There are a number of different formulae for estimating the variogram. Each method has different strengths, weaknesses, and proponents. **Surfer** contains four estimating options: the *Variogram*, *Standardized Variogram*, *Autocovariance*, and *Autocorrelation*. When in doubt use the default classical *variogram*.

An observation is an (X,Y,Z) triple. A pair of observations contains two such triples. Consider the set of N pairs of observations that go into the calculation of a single plotted point on an experimental variogram. These N pairs are ordered pairs; the two observations are not randomly ordered within a pair. The first observation in the pair is the point with the smaller Y value and the second observation is the point with the larger Y value. The first observation is called the "head" (denoted by an "h"), and the second observation is called the "tail" (denoted by a "t").

Let

Zh_i be the observed value of the head observation of the i^{th} pair, and

Zt_i be the observed value of the tail observation of the i^{th} pair.

The required sums over the set of all N pairs are denoted:

$$Zh = \sum Zh_i$$

$$Zhh = \sum Zh_i^2$$

$$Zt = \sum Zt_i$$

$$Ztt = \sum Zt_i^2$$

$$Zht = \sum Zh_i \times Zt_i$$

The average and variance for all of the observed head values are:

$$AVEh = \frac{Zh}{N}$$

$$VARh = \frac{Zhh}{N} - (AVEh)^2$$

Similarly, the average and variance for all of the observed tail values are:

$$AVE_t = \frac{Z_t}{N}$$

$$VAR_t = \frac{Z_{tt}}{N} - (AVE_t)^2$$

Then define the following two intermediate quantities:

$$\text{Lag Variance} = \sqrt{VAR_h \times VAR_t}$$

$$\text{Lag Covariance} = \left(\frac{Z_{ht}}{N} \right) - (AVE_h \times AVE_t)$$

Variogram

The classical variogram estimator is:

$$\gamma = \frac{(Z_{hh} - 2Z_{ht} + Z_{tt})}{2N}$$

(see Pannatier, 1996, p. 38).

Standardized Variogram

The standardized variogram estimator is:

$$\gamma = \frac{\frac{(Z_{hh} - 2Z_{ht} + Z_{tt})}{(2N)}}{\text{Lag Variance}}$$

(see Pannatier, 1996, p. 39).

Autocovariance

The autocovariance estimator is:

$$\gamma = \text{Lag Variance} - \text{Lag Covariance}$$

(see Pannatier, 1996, p. 41).

Chapter 5 - Variograms

Autocorrelation

The autocorrelation estimator is:

$$\gamma = \frac{1 - \text{Lag Covariance}}{\text{Lag Variance}}$$

(see Pannatier, 1996, p. 42).

Maximum Lag Distance

The *Max Lag Dist* specifies the extent of the separation distance axis (abscissa or X axis). This value should never be larger than the *Max Lag Distance* specified in the **New Variogram** dialog. Pairs with separation values beyond the **New Variogram** *Max Lag Distance* have been eliminated from the variogram already. The *Max Lag Distance* specified when creating the variogram grid (**New Variogram** dialog) limits the maximum separation distance that you can look without recreating a new variogram grid. The *Max Lag Dist* specified in the variogram properties dialog limits the maximum separation distance you will look. The *Max Lag Dist* specified in the **Experimental** page in the variogram properties dialog is often less than the *Max Lag Distance* specified when creating the variogram grid.

Number of Lags

The *Number of Lags* specifies how many experimental variogram points are computed and plotted. The default *Number of Lags* is 25.

Lag Width

When the associated *Auto* check box is checked, the *Lag Width* is automatically set equal to the *Max Lag Dist* divided by the *Number of Lags*. For example, if the *Max Lag Dist* is 100, and the *Number of Lags* is five, the *Lag Width* would be set to $100 / 5 = 20$. In this case, the first plotted point would summarize all pairs with separation distances greater than or equal to zero and less than 20. The second plotted point would summarize all pairs with separation distances greater than or equal to 20 and less than 40, etc. Lag intervals are 0 to 20, 20 to 40, ..., 80 to 100; they do not overlap. You can use *Lag Width* to smooth a variogram.

Smoothing the Variogram with Lag Width

When the associated *Auto* check box is not checked, the *Lag Width* may be independently specified. In this case, the *Lag Width* is not a function of the *Max Lag Dist* and the *Number of Lags*. This flexibility allows for overlapping lag intervals and, therefore, a smoothed experimental variogram.

For example, consider the case where the *Max Lag Dist* is 100, the *Number of Lags* is five, and the *Lag Width* is 40. The resulting lag intervals would be 0 to 40, 10 to 50, 30 to 70, 50 to 90, and 60 to 100. Notice these intervals overlap. Overlapping lag intervals causes a moving average type smoothing of the experimental variogram. Often, it is easier to select an appropriate model for such smoothed experimental variograms.

In either case, the formulae for determining the intervals are given by the following pseudo-code fragment.

```
for(lag = 0; lag < Number of Lags; ++lag )
{
    center = Max Lag Distance / Number of Lags * (lag+0.5);
    from   = max( center - ( Lag Width / 2.0 ), 0.0 );
    to     = min( center + ( Lag Width / 2.0 ), Max Lag Distance );
}
```

The screenshot shows a software dialog box titled "Experimental". It is divided into two main columns. The left column, under the heading "Lag Direction", contains four input fields: "Direction" with the value 15, "Tolerance" with the value 30, "Step Amount" with the value 30, and a circular diagram. The circular diagram has a blue sector in the lower-left quadrant, representing an angular window. Below the diagram are two buttons: "Step CW" and "Step CCW". The right column contains five input fields: "Estimator Type" is a dropdown menu set to "Variogram"; "Max Lag Dist" is 180; "Number of Lags" is 25; "Lag Width" is 7.2 with an "Auto" checkbox checked; and "Vertical Scale" is 2600 with an "Auto" checkbox checked.

*The **Experimental** page in the variogram properties dialog contains the parameters for the experimental variogram. The diagram in the Lag Direction group shows the direction and size of the angular window of the current variogram.*

Chapter 5 - Variograms

Vertical Scale

The *Vertical Scale* specifies the extent of the variogram axis (ordinate or Y axis). If the associated *Auto* check box is checked, the vertical scale is determined on the fly so that the entire variogram fits on the plot. When the *Auto* check box is checked the vertical scale is recomputed every time the variogram is redrawn. As you step around with the *Step CCW* or *Step CW* buttons, the vertical scale changes with each step.

When the associated *Auto* check box is not checked, the *Vertical Scale* is held constant at the specified value. When using the *Step CCW* and *Step CW* buttons to rotate the lag direction, it is advisable to uncheck the *Auto* check box to avoid having the vertical scale change for every plot. This allows for easier visual comparison between plots.

Model

The **Model** page in the variogram properties dialog allows you to set a specific variogram model and its parameters. Computing an experimental variogram from your data is the only certain way to determine which variogram model you should use. A detailed variogram analysis can offer insights into the data that would not otherwise be available, and it allows for an objective assessment of the variogram scale and anisotropy. There are lengthy chapters in many geostatistics textbooks discussing the tools and techniques necessary to generate a variogram (e.g. Isaaks and Srivastava, 1989).

Surfer allows for a general nested variogram model. There are no limits on the number of models that can be nested in **Surfer**. Because of this, there are hundreds of possible combinations of variogram models. There are twelve common variogram functions: *Nugget Effect*, *Spherical*, *Exponential*, *Linear*, *Gaussian*, *Wave (Hole-Effect)*, *Quadratic*, *Rational Quadratic*, *Logarithmic*, *Power*, *Cubic*, and *Pentasppherical*. Each of the components allow for independent specification of the anisotropy, except for the nugget effect.

When in doubt, you should use the *Linear* variogram model with the default *Scale (C)* and *Length (A)* parameters.

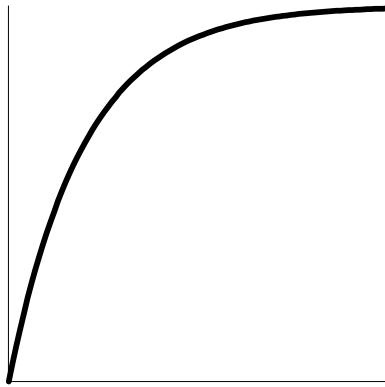
The **variogram model** components available in **Surfer** are:

equation variables:

$\gamma(h)$ is the semi-variogram

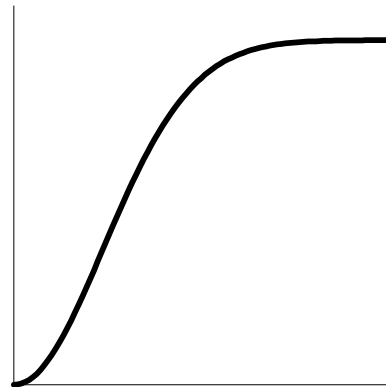
C is the scale for the structured component of the variogram

h is the anisotropically



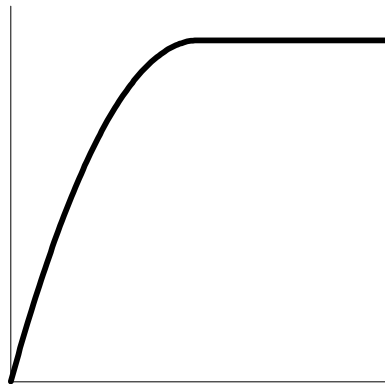
Exponential Model
Cressie (1991, p. 61)

$$\gamma(h) = C[1 - e^{-h}]$$



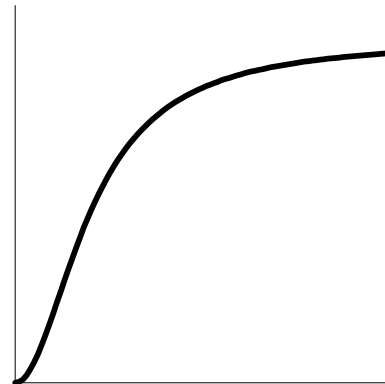
Gaussian Model
Pannatier (1996, p. 50)

$$\gamma(h) = C\left[1 - e^{-h^2}\right]$$



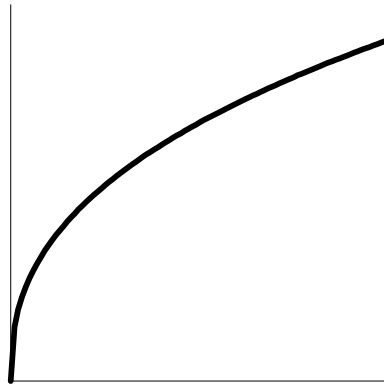
Quadratic Model
Alfaro (1980, p. 31)

$$\gamma(h) = \begin{cases} C[2h - h^2] & h < 1 \\ C & h \geq 1 \end{cases}$$

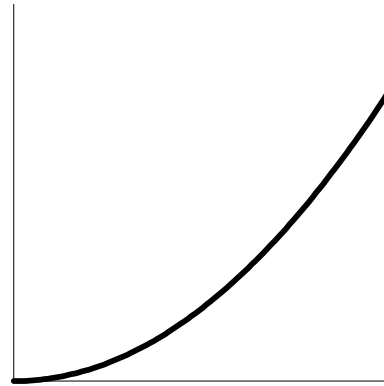


Rational Quadratic Model
Cressie (1991, p. 61)

$$\gamma(h) = C\left[\frac{h^2}{1 + h^2}\right]$$



$$0 < h < 1$$



$$1 < h < 2$$

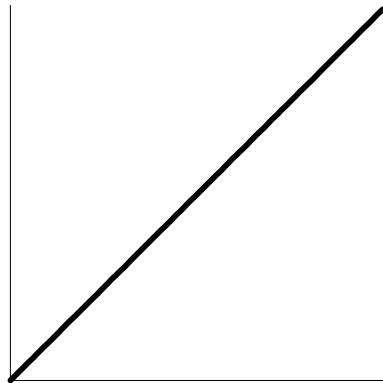
Power Models

Pannatier (1996, p. 51)

$$\gamma(h) = C[h^n]$$

where $0 < n < 2$

when $n=1$ it is a linear model

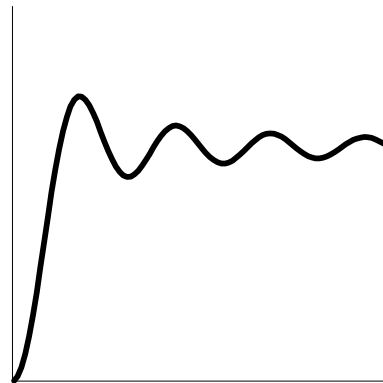


Linear Model

Kitanidis (1997, p. 61)

$$\gamma(h) = C(h)$$

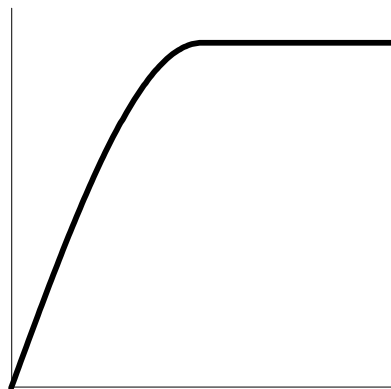
The Linear Model is a special case of the Power Model where the power is equal to one.



Wave (Hole Effect) Model

Cressie (1991, p. 62)

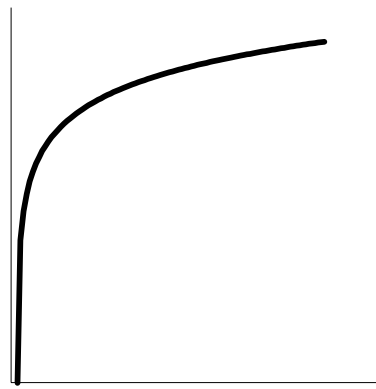
$$\gamma(h) = C \left[1 - \frac{\sin h}{h} \right]$$



Spherical Model

Pannatier (1996, p. 48)

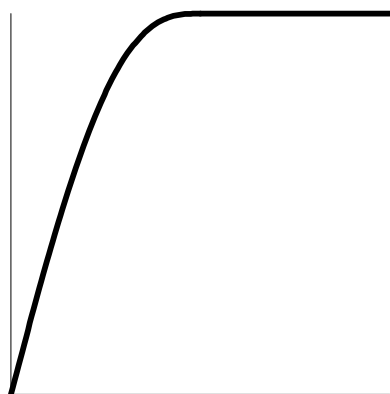
$$\gamma(h) = \begin{cases} C[1.5h - 0.5h^3] & h < 1 \\ C & h \geq 1 \end{cases}$$



Logarithmic Model

Kitanidis (1997, p. 61)

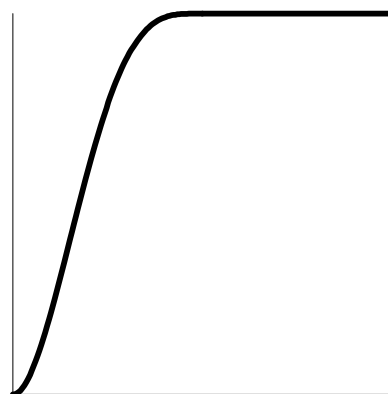
$$\gamma(h) = C[\log_e(h)] \quad h > 0$$



Pentaspherical Model

Olea (1999, p. 79)

$$\gamma(h) = C(1.875h - 1.25h^3 + 0.375h^5)$$

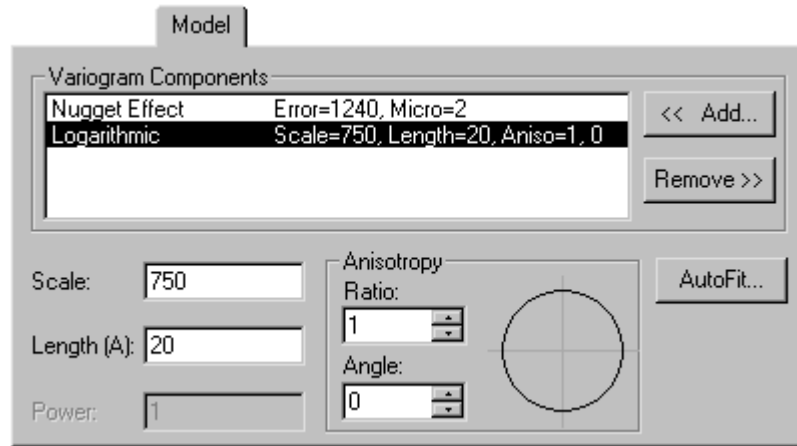


Cubic Model

Olea (1999, p. 79)

$$\gamma(h) = C(7h^2 - 8.75h^3 + 3.5h^5 - 0.75h^7)$$

Chapter 5 - Variograms



Options such as the variogram model are set through the **Model** page.

With the exception of the *Linear*, *Logarithmic*, and *Power* variogram models (which do not have a sill), the *Scale* parameters (denoted by C in the variogram equations) define the sill for the variogram components you select. Thus, the sill of the variogram model equals the *Nugget Effect* plus the sum of the component's *Scale* (C) parameters. In many situations, the variogram model sill is approximately equal to the variance of the observed data (see Barnes, 1991).

The *Length* (A) parameters define how rapidly the variogram components change with increasing separation distance. The *Length* (A) parameter for a variogram component is used to scale the physical separation distance. For the *Spherical* and *Quadratic* variogram functions, the *Length* (A) parameter is also known as the variogram range. In an isotropic setting, the relative separation distance, h , is computed by the following equation:

$$h = \frac{\sqrt{\Delta x^2 + \Delta y^2}}{A}$$

where

$[\Delta x \ \Delta y]$ is the separation vector (in map coordinates), and
 A is the component's *Length* parameter.

The anisotropically rescaled relative separation distance for the variogram equations is computed by the following matrix equation

$$h = \sqrt{[\Delta x \quad \Delta y] \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} (1/A)^2 & 0 \\ 0 & (\rho/A)^2 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}}$$

where

$[\Delta x \quad \Delta y]$ is the separation vector (in map coordinates),

A is the component's *Length* parameter.

θ is the anisotropy *Angle*, and

ρ is the anisotropy *Ratio*.

With a *Linear* variogram model, the slope is given by the *Scale* divided by the *Radius*. By allowing an anisotropic radius, it is possible to specify an anisotropic linear variogram slope. The slope of the linear variogram makes no difference in the kriging algorithm if the nugget effect is zero.

Nugget Effect

The *Nugget Effect* is used when there are potential errors in the collection of your data. The nugget effect is implied from the variogram you generate of your data. Specifying a nugget effect causes kriging to become more of a smoothing interpolator, implying less confidence in individual data points versus the overall trend of the data. The higher the nugget effect, the smoother the resulting grid. The units of the nugget effect are the units of the observations squared.

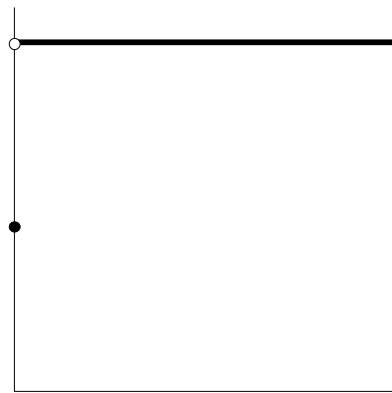
The nugget effect is made up of two components:

$$\text{Nugget Effect} = \text{Error Variance} + \text{Micro Variance}$$

Chapter 5 - Variograms

The *Error Variance* is a measure of the direct repeatability of the data measurements. If you took an observation at one point, would the second measurement at that exact point be exactly the same as the first measurement? The error variance values take these variances in measurement into account. A non-zero error variance means that a particular observed value is not necessarily the exact value of the location. Consequently, kriging tends to smooth the surface: it is not a perfect interpolator.

The *Micro Variance* is a measure of variation that occurs at separation distances of less than the typical nearest neighbor sample spacing. For example, consider a nested variogram where both of the models are spherical. The range of one of the structures is 100 meters while the range of the second structure is five meters. If our closest sample spacing were 10 meters, we would not be able to see the second structure (five meter structure). The micro variance box allows you to specify the variance of the small-scale structure. If you do not know the micro variance, leave the setting at 0.0.



Nugget Effect Model

$$\gamma(h) = \begin{cases} \text{Error Variance} & h = 0 \\ \text{Error Variance} + \text{Micro Variance} & h > 0 \end{cases}$$

This is the nugget effect model. The rest of the model graphics are located in the beginning of the Model section of this chapter.

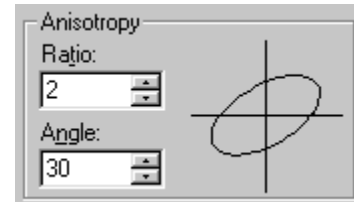
Surfer's implementation of the nugget effect follows the recommendation of Cressie (1991, Section 3.2.1), and this reference should be consulted for a more rigorous presentation of the nugget effect and its partitioning.

Anisotropy

Physical processes have preferred orientations. Anisotropy implies a preferred direction, or direction of higher or lower continuity between data points. Alternatively, you can say that points that lie farther away along one axis are given equivalent weights to points that lie closer along the other axis. The relative weighting is defined by the anisotropy ratio.

An example of when anisotropy might be employed is an isotherm map (contour map) of average daily temperature over the upper Midwest. Although the X and Y coordinates (i.e., Easting and Northing) are measured using the same units, along the east-west lines the temperature tends to be very similar. Along north-south lines, the temperature tends to change more quickly (getting

colder as you head north). When creating a variogram with this data it would be advantageous to give more weight to data along the east-west axis than along the north-south axis. When interpolating a grid node, observations that lie in an east-west direction are given greater weight than observations lying an equivalent distance in the north-south direction.



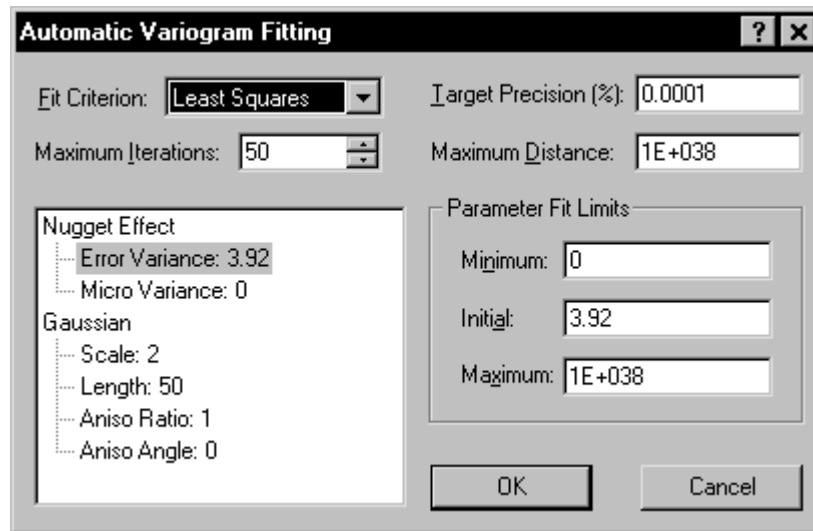
Anisotropy is also useful when data sets use fundamentally different units in the X and Y dimensions. For example, consider plotting a flood profile along a river. The X coordinates are locations, measured in miles along the river channel. The Y coordinates are time, measured in days. The Z values are river depth as a function of location and time. Clearly in this case, the X and Y coordinates would not be plotted on a common scale, because one is distance and the other is time. One unit of X does not equal one unit of Y. While the resulting map can be displayed with changes in scaling (see *Chapter 14, Common Map Properties*), it may be necessary to apply anisotropy as well.

*The anisotropy settings are located on the **Model** page in the variogram properties dialog.*

The anisotropy settings in the **Model** page in the variogram properties dialog include a *Ratio* and an *Angle* setting. The ratio is the primary range divided by the secondary range. An anisotropy ratio less than two is considered mild, while an anisotropy ratio greater than four is considered severe. Typically, when the anisotropy ratio is greater than three its effect is clearly visible on contour maps. The *Angle* is the orientation (direction) of the primary axis.

AutoFit

AutoFit takes a user specified variogram model and an initial set of parameters, and attempts to find a better set of parameter values. For example, if a user has selected a spherical variogram model with a nugget effect, the AutoFit tool can be used to determine the "best" fitting anisotropy parameters.



*The Automatic Variogram Fitting dialog is opened by clicking the AutoFit button on the **Model** page of the variogram properties dialog. The AutoFit feature does not select an appropriate model. Before using AutoFit, select reasonable model parameters and then use AutoFit to fine tune the model.*

The *AutoFit* button on the **Model** page in the variogram properties dialog takes the user into the **Automatic Variogram Fitting** dialog. Please note that the AutoFit tool is meant to hone and polish a set of user specified parameters. **AutoFit does not select the appropriate model form.** It should not be used as an exploratory tool in the search for an appropriate model. It merely fits the parameters *once you select a reasonable model*.

The underlying fitting algorithm is a simple implementation of Powell's Search Method. This search does not involve the use of derivatives. This means there are no gradient calculations. One iteration of this algorithm includes a line search along each of the individual parameter directions, and then a line search along the accumulated direction moved during this iteration. There is some useful discussion of this method in Press et al. (1988, Section 10.1). However, this routine does not attempt to build a set of conjugate directions.

While the fitting algorithm is working, the current iteration number and the current value of the objective function are indicated in the status bar. The fit algorithm can take some time to work.

Fit Criterion

There are two fit criteria available, *Least Squares* and *Least Absolute Value*. An important consideration is that both of these criteria work by fitting the model to the variogram grid in all directions at once and not to the current directional variogram plot shown on the screen.

The *Least Squares* criterion is an implementation of the simple weighted least squares method discussed in Cressie (1991, Section 2.6.2). See Zimmerman and Zimmerman (1991) for a rather detailed comparison among various methods. The *Least Squares* method minimizes the sum of the square of the error, and the *Least Absolute Value* method minimizes the sum of the absolute value of the errors.

The *Default* method uses the default linear variogram. Default calculates a slope and nugget effect similar to the default linear variogram as first displayed with **New Variogram**.

Maximum Iterations

The maximum number of iterations specifies a termination criterion for AutoFit. After this number of iterations the search stops, though the iterations also terminate if a target precision is met (see the next section). One iteration includes a line search for each of the free parameters, plus a line search in the direction of the accumulated change.

One nice feature is that the AutoFit search can be stopped and restarted multiple times. Therefore, you could run the search for a few iterations, see what is happening, and then try a few more iterations. You can terminate the search by clicking the ESC button on your keyboard and selecting the *Abort* button in the dialog that appears. If you stop AutoFit, the best set of parameters so far is saved.

Target Precision

The target precision is also a termination criterion for the search with AutoFit. The target precision specifies a maximum relative change in the objective function during one iteration. If the change in the objective function is less than the specified amount, the search is terminated. If the change in the objective function is greater than the specified amount, the search is continued for another iteration (assuming that the maximum number of iterations has not been reached). The actual formula is

$$\text{OldObjective} - \text{NewObjective} < \text{OldObjective} * \text{TargetPrecision}/100$$

Note that the target precision is given as a percentage.

Maximum Distance

AutoFit attempts to fit the model to the data contained in the variogram grid. The extent of the variogram grid is given by the *Max Lag Distance* parameter specified when the variogram grid is created with the **New Variogram** dialog. Often, the *Max Lag Distance* specified when the variogram grid is created is greater than the maximum lag distance that a user is interested in working with while fitting. (Remember that the variogram grid is created before you start modeling).

The *Maximum Distance* parameter limits the extent of the search. The fitting algorithm ignores any pairs separated by more than the *Maximum Distance*, even if they are included in the variogram grid.

Parameter Fit Limits

The *Parameter Fit Limits* define the range of values over which the fitting algorithm searches. The *Parameter Fit Limits* allow the user a great deal of control in applying the fitting algorithm. For example, if you specify the *Maximum* equal to the *Minimum*, the fitting algorithm does not adjust the specified parameter.

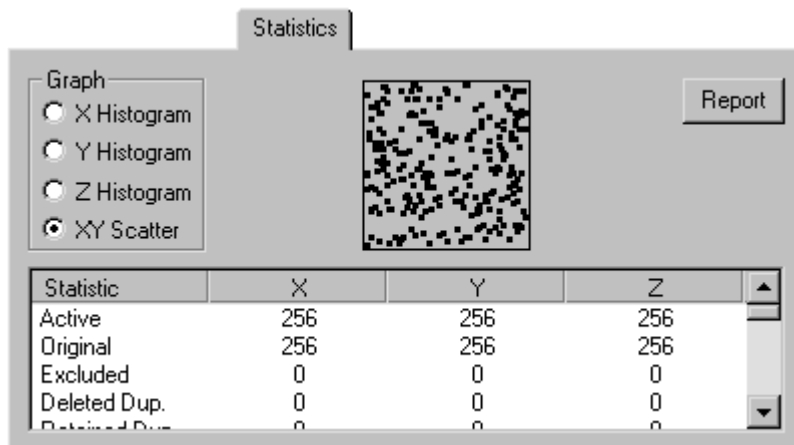
Although the fit algorithm attempts to fit parameters with a large range between the *Maximum* and the *Minimum*, you will have greater success with the fitting algorithm if the limits are as tight as you can make them. Use a "reasonable value" approach in specifying the limits.

- The *Minimum* is the lower bound on the fitted parameter value. No value lower than this is considered by the fitting algorithm.
- The *Initial* option is a guess for the parameter value. Since the objective function is non-linear, the final fitted values can be sensitive to the starting point. The objective function (fit criterion) may have numerous local minima, which are distinct from the global minima.
- The *Maximum* is the upper bound on the fitted parameter value. No value greater than this is considered by the fitting algorithm.

Statistics

The **Statistics** page in the variogram properties dialog shows histograms and scatter plots of the data, as well as providing statistics on the data.

The *X, Y, Z Histogram* options show the distribution of the X, Y, and Z data respectively. The histogram is shown to the right of the options. The *XY Scatter* option shows the distribution of the data as a simple scatter diagram.



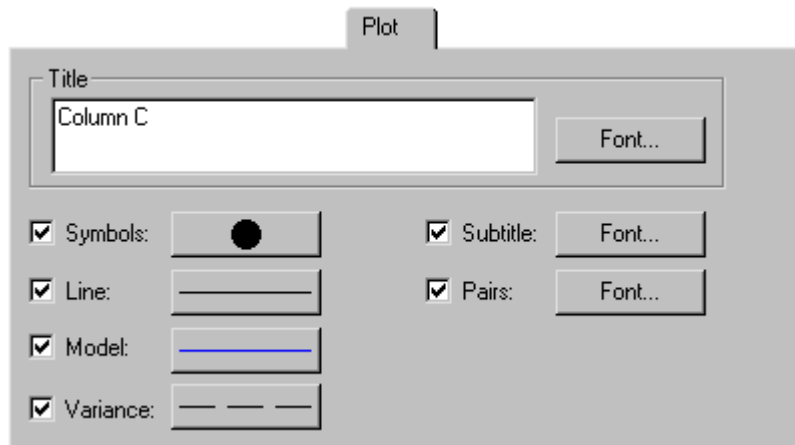
*On the **Statistics** page, you can show an X, Y, or Z histogram, view statistics, or generate a variogram report.*

You can use the scroll bar on the right of the statistics box to view the statistics. Click the *Report* button to open the statistics and variogram information in a report window. The statistics can be printed if they are opened in the report window.

Plot

Most of the visual characteristics of the variogram display are controlled through the **Plot** page.

- You can change the title and title font in the *Title* group. To create a multi-lined title, press CTRL+ENTER on your keyboard to move to the next line. See the online help system for more information on the **Text Properties** dialog.
- To display *Symbols*, the plot *Line*, the *Model* line, the *Variance* line, the direction/tolerance *Subtitle*, or the number of *Pairs*, click in the check box to the left of the option name.
- To change the characteristics of the feature, click the button to the right of the option name and make the changes in the dialog.



*You can control the display of the variogram graph through the **Plot** page in the variogram properties dialog.*

Default Linear Variogram

The default linear variogram takes the form

$$\gamma(h) = C_0 + S \cdot h$$

where C_0 is the unknown nugget effect, and S is the unknown slope. We need two defining equations to solve for these two unknown parameters.

According to theory, the expected value of the sample variance is the average value of the variogram between all possible pairs of sample locations (Barnes, 1991); this yields one equation. The second equation is generated by equating the experimental sample variogram for nearest neighbors to the modeled variogram. Thus, we can write

$$Var = C_0 + S \cdot D_{ave}$$

$$G_{nn} = C_0 + S \cdot D_{nn}$$

where

D_{nn} = average distance to the nearest neighbor

D_{ave} = average inter-sample separation distance

G_{nn} = one half the averaged squared difference between nearest neighbors

Var = sample variance

By solving the two equations for the two unknown parameters, and checking for unreasonable values, we get the final formulae used in **Surfer**:

$$S = \max \left[\frac{Var - G_{nn}}{D_{ave} - D_{nn}}, 0 \right]$$

$$C_0 = \max \left[\frac{G_{nn} \cdot D_{ave} - Var \cdot D_{nn}}{D_{ave} - D_{nn}}, 0 \right]$$

Exporting a Variogram

You can use **Grid | Variogram | Export Variogram** to export the XY coordinates of the symbols representing the pairs for each lag distance. The resulting file is an ASCII text file ([.DAT] extension), with the X coordinates (separation distance) in Column A, the Y coordinates (variogram - $\gamma(h)$) in Column B, and the number of pairs in Column C.

Using Variogram Results in Kriging

Once you have completed the variogram, you can use the results of the variogram model when gridding the data with the *Kriging* gridding option.

To use model results:

1. Make sure the plot window containing the variogram is open. If you switch to a new plot window, the variogram results cannot be used.
2. Select **Grid | Data**.
3. In the **Open** dialog, choose the data file to be modeled and then click the *Open* button.
4. In the **Grid Data** dialog, select *Kriging* for the gridding method.
5. Click the *Advanced Options* button to open the **Kriging Advanced Options** dialog.
6. Click the *Get Variogram* button to use the variogram model open in the plot window. If more than one variogram exists in the document, a dialog is displayed prompting you for the variogram to use.

For more information about kriging, see the *Chapter 4, Creating Grid Files*.

Suggested Reading

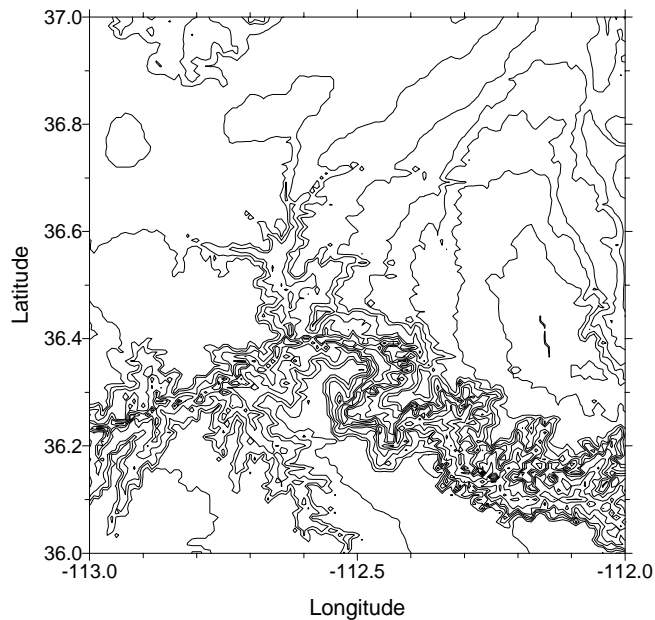
- Alfaro, M. (1980), The Random Coin Method: Solution of the Problem of the Simulation of a Random Function in the Plane, *Mathematical Geology*, v. 12, n. 1, p. 25-32.
- Barnes, R. (1991), The Variogram Sill and the Sample Variance, *Mathematical Geology*. v. 23, n. 4, p. 673-678.
- Cressie, Noel (1991), *Statistics for Spatial Data*, John Wiley and Sons, New York, 900 pp., ISBN 0-471-84336-9.
- Isaaks, Edward and Mohan Srivastava (1989), *An Introduction to Applied Geostatistics*, Oxford University Press, New York, 561 pp., ISBN 0-19-505013-4.
- Kitanidis, Peter (1997), *Introduction to Geostatistics - Applications in Hydrogeology*, Cambridge University Press, New York, 249 pp., ISBN 0-521-58747-6.
- Olea, R.A., 1999, *Geostatistics for Engineers and Earth Scientists*, Kluwer Academic Publishers, Boston, 303 pp. ISBN 0-7923-8523-3.
- Pannatier, Yvan (1996), *VarioWin - Software for Spatial Data Analysis in 2D*, Springer-Verlag, New York, 91 pp., ISBN 0-387-94579-9.
- Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. (1988), *Numerical Recipes in C*, Cambridge University Press.
- Zimmerman, D. L., and M. B. Zimmerman (1991), A Comparison of Spatial Semivariogram Estimators and Corresponding Ordinary Kriging Predictors, *Technometrics*, v. 33, n. 1, p. 77-91.

Chapter 6

Contour Maps

Introduction to Contour Maps

The **Map | Contour Map | New Contour Map** command creates a contour map from a **Surfer** grid file [.GRD], USGS Digital Elevation Model ([.DEM], [.HDR], or [.DDF]) or Digital Terrain Elevation Data (DTED) [.DT*]. A contour map is a two-dimensional representation of three-dimensional data. The first two dimensions are the XY coordinates, and the third dimension (Z) is represented by lines of equal value. The relative spacing of the contour lines indicate the relative slope of the surface. The area between two contour lines contains only grid nodes having Z values within the limits defined by the two enclosing contours. The difference between two contour lines is defined as the contour interval.



This contour map shows a portion of the Grand Canyon.

As an example, consider an island in a lake viewed from above. The island shoreline represents the same elevation (the same Z value) around the entire island. Viewed from above, the shoreline can be thought of as a contour line. If the level of the lake drops by two feet, the new shoreline represents a new contour line. If the water continues to drop in two-foot intervals, additional contours would be generated, each accurately expressing the shoreline at each elevation (each Z level). A contour map created in this way would display contours at two-foot intervals, describing the island's shoreline at different elevations. An excellent introduction to contour maps is given in the free booklet, "Topographic Mapping," issued by the U.S. Geological Survey.

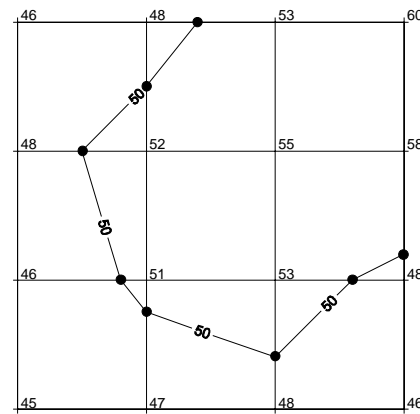
Chapter 6 - Contour Maps

Drawing Contours

When you create a contour map, the grid file is read into the plot window as an internal array of XYZ grid nodes. The grid nodes consist of rows and columns of Z values. The rows contain grid nodes with the same Y coordinate, and the columns contain grid nodes with the same X coordinate. Grid file columns and rows are sometimes referred to as X grid lines and Y grid lines, respectively. The intersection of a row and column is defined as a grid node. Grid files define the XY location of each grid node over the extent of the map, and the interpolated Z value at each node.

The grid limits define the extent of contour maps. Once a grid file is created you cannot produce a contour map larger than the extent of the grid file. However, you can use the map properties **Limits** page to specify a subset of the grid used to create the contour map. If a larger area is needed, the original data file will need to be regridded to contain the larger limits. This is done with the **Grid | Data** command.

The elevation data contained in USGS DEM files are regularly spaced, but the direction of the grid lines may not exactly coincide with the grid edges. This is due to the variable angle between the UTM coordinate system (used in some DEM files) and true north. This results in the potential for a regular, stair-stepped arrangement of the grid nodes along the edges of the map that can translate to blanked grid nodes along the edges of maps produced in **Surfer**.



The path of a contour line ($Z = 50$) through a grid is defined by interpolating between grid nodes. This example shows the straight-line segments that define the contour line. The dots are added to show the ends of each line segment.

When **Surfer** creates a contour map, the contour lines are drawn as a series of straight-line segments between adjacent grid lines. The point where a contour line intersects a grid line is determined by interpolation between Z values at adjacent grid nodes.

Contour Map Components

Contour maps consist of several different components, each of which can be customized to control the appearance of the map. The components are described briefly here.

- Contour levels define which contour lines appear on the map, and the contour interval between adjacent contour lines.
- Color fill between contours can consist of a gradually changing spectrum of colors, or color fill can be independently defined for each contour level.
- Contour line properties include the color, style, and thickness of the contour lines. The contour lines on a map can gradually change their color or thickness (or both) from the highest to the lowest contour. Each level's contour line properties can also be independently specified.
- Contour labels are numbers on a contour line that indicate the Z value of the contour line.
- Contour smoothness controls the roundness of the contour lines. Smoothness in a contour map can be controlled by:
 1. selecting the *Smoothing* option in the contour map properties dialog,
 2. using the **Grid | Spline Smooth** command,
 3. using the **Grid | Filter** command,
 4. producing a denser grid file from your original data,
 5. or changing the gridding method or gridding method parameters that control the gridding process.

Some methods produce smoother grids than others, and some have smoothing parameters that can be utilized during gridding.


- Hachures are tick marks along the contour lines that indicate upward or downward slopes.
- Map scale controls the size of the map and the number of map units per page unit.

Chapter 6 - Contour Maps

- Map axes display tick marks and axis labels along the borders of the map. Axes show the X and Y scaling in map units. You can include grid lines as extensions of the tick marks over the extent of the map.
- Map orientation is defined as the rotation, tilt, and field of view of the contour map.
- Map limits are the XY extents of the contour map.
- A background can be added to a map. The map background consists of a fill underneath the map and a border around the map.

Creating a New Contour Map

To create a new contour map:

1. Click **Map | Contour Map | New Contour Map** or click the  button.
2. Select a grid file in the **Open Grid** dialog and click the *Open* button.
3. The contour map is created.

Editing Contour Map Properties

To change the features of the contour map, open the contour map properties dialog by:

- double-clicking on the map in the plot window,
- double-clicking on the map name in the **Object Manager**,
- right-clicking the map and choosing **Properties** in the context menu,
- or by clicking **Edit | Properties** when the contour map is selected.

Contour Map Properties

The contour map properties dialog contains six pages: **General**, **Levels**, **View**, **Scale**, **Limits**, and **Background**. **View**, **Scale**, **Limits**, and **Background** are discussed in *Chapter 14, Common Map Properties*.


General

The contour map properties **General** page contains the following options:


Input Grid File

The *Input Grid File* lists the current grid file used in the contour map.



The  button displays information about the file used to produce the contour map. The information includes the grid size, the minimum and maximum XYZ values contained in the grid file, and statistics. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.



The  button displays the **Open Grid** dialog. This allows you to open a new file, or an updated version of the file used to create the contour map. When the grid file is changed, the map limits are reset but the contour levels are not recalculated. If the Z range for the new grid is outside the old contour levels, no contour lines are redrawn until the contour levels are reset to match the new grid.

Filled Contours

The *Filled Contours* group controls the display of color fill between the contour lines as well as the display of the color scale bar. See the *Color Fill* section in this chapter for more information on filled contours.

Smoothing

The *Smoothing* group allows you to smooth the contours displayed on the map. See the *Smoothing Contours* section of this chapter for more information on smoothing.

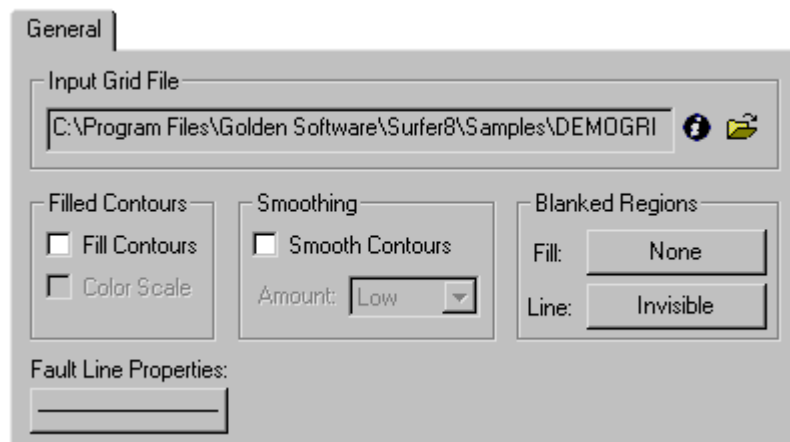
Chapter 6 - Contour Maps

Blanked Regions

The *Blanked Regions* group allows you to choose fill and line color properties for areas containing blanked nodes. Click on the *Fill* button to choose a fill pattern and fill color. Click the *Line* button to outline the blanked area with a specific line style, color, and width.

Faults

If your grid file contains faulting information, you can set the fault line color, style, and thickness of the fault line drawn on the contour map by clicking on the *Fault Line Properties* button.

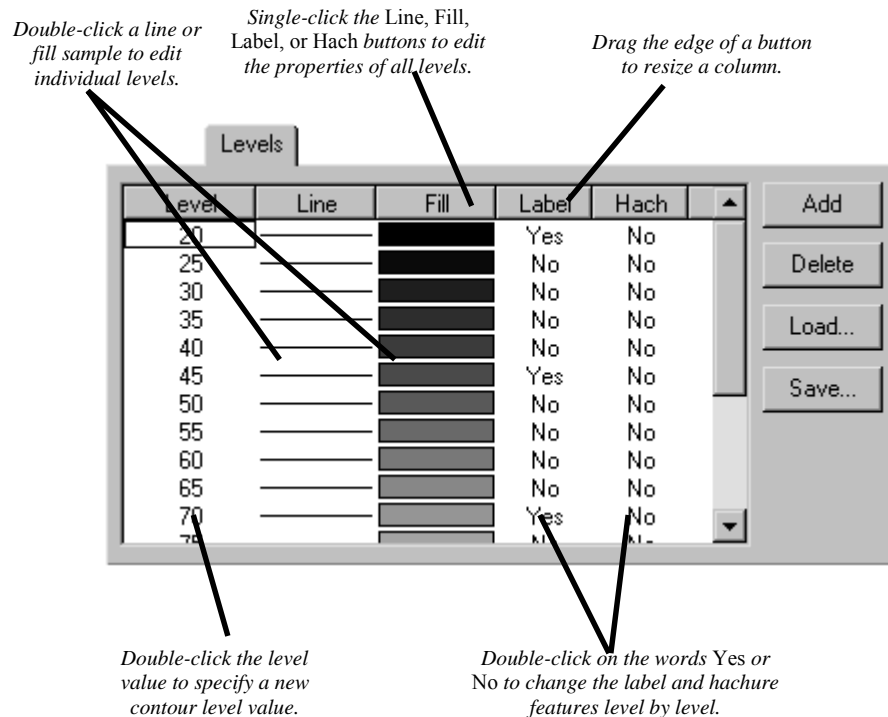


The **General** page in the contour map properties dialog allows you to control the grid file, contour fill, smoothing, blanked area properties, and fault line properties.

Levels

The **Levels** page in the contour map properties dialog controls the display of contours, contour labels, color fill, and hachures on the contour map. The contour level list shows the contour levels to be displayed on the map. Contour level parameters can be set as a group or individually.

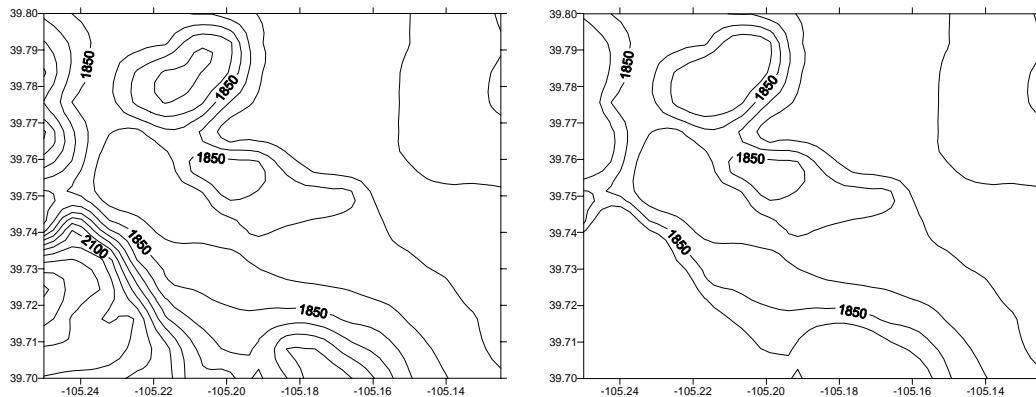
- To apply level parameters to all defined levels, click the *Level*, *Line*, *Fill*, *Label*, or *Hach* buttons at the top of the contour level list. For example, to define the contour range and the interval between contours, click the *Level* button and enter the values you would like to use.
- To set parameters for an individual contour level, double-click any of the elements in the list.
- The *Load* and *Save* buttons allow you to load and save level files [.LVL] containing contour level settings.
- Use the *Add* and *Delete* buttons to add or delete specific contour levels.



Chapter 6 - Contour Maps

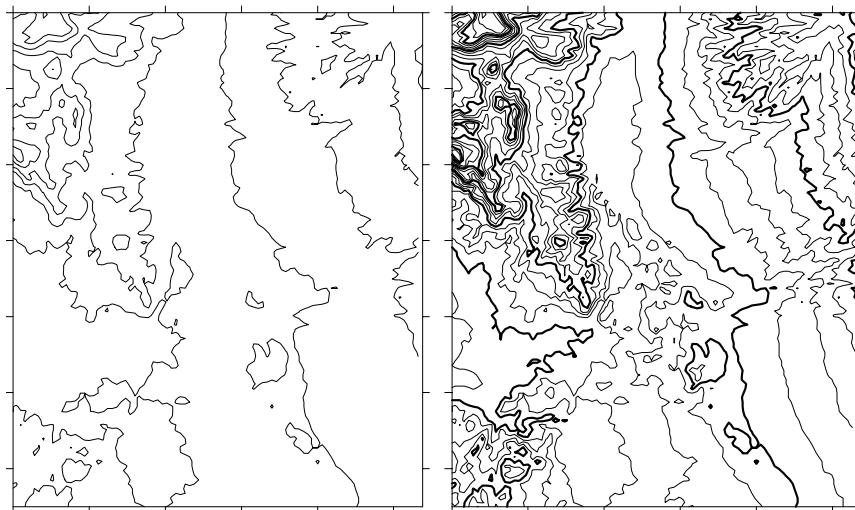
Contour Levels

The **Contour Levels** dialog is used to assign regularly spaced contour levels. For example, assume the smallest Z value is defined as 50, the largest Z value is 100, and the interval between contours or zones is 10. In this case, contour lines are drawn at Z levels of 50, 60, 70, 80, 90, and 100. If the range of grid Z values goes beyond the defined minimum and maximum contour levels, the contours outside the levels are not drawn.



When the contour range does not extend to the Z limits in the grid file, the contour levels outside the range are not drawn. In the contour map on the right, the maximum contour level is set below the maximum Z value in the grid file, and as a result, the highest contour lines are not drawn.

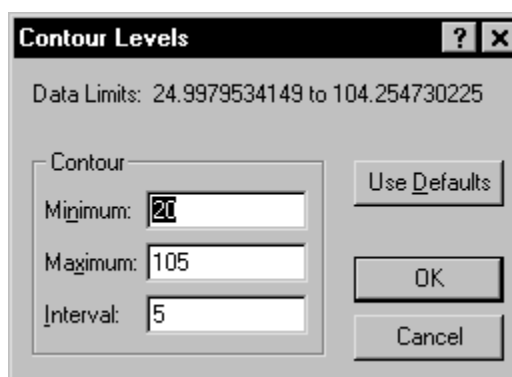
Different contour intervals provide different amounts of detail to a contour map. In the examples shown on the next page, two different contour intervals are used for the same grid file. The contour interval for the map on the left is much larger than the interval used for the map on the right. Consequently, the map on the right has more contour lines, and provides more detail about the grid file than the map on the left.



Different contour intervals show different amounts of detail about the grid file used to produce the map.

To create contour levels at regular intervals:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Click the **Levels** page.
3. To assign evenly spaced contour levels, click the *Level* button and the **Contour Levels** dialog is displayed.
4. Enter values for the *Minimum* and *Maximum* contours, and the *Interval* value that defines the spacing in Z units between adjacent contour lines.
5. The *Use Defaults* button can be used to calculate a reasonable set of parameters based on the range of Z values within the grid.
6. Click the OK button and the contour level list is updated.
7. Click OK or *Apply* in the map properties dialog to redraw the map.

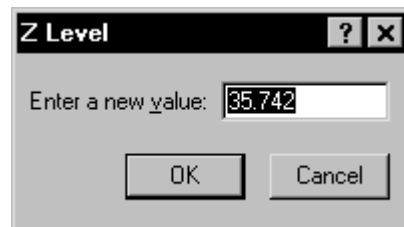


*Open the **Contour Levels** dialog by clicking the *Level* button in the map properties dialog.*

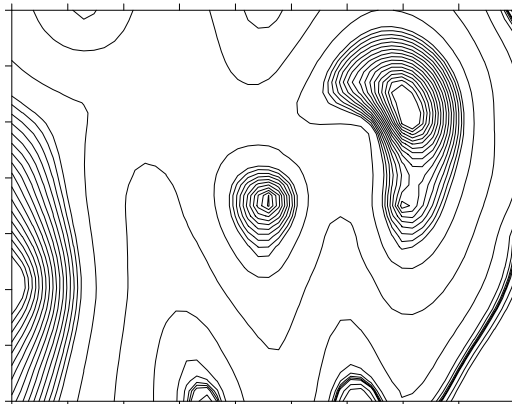
Chapter 6 - Contour Maps

To add, delete, or modify individual contour levels:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Click the **Levels** page.
3. Click the *Add* button to add a contour level midway between the selected level and the next lower level. Adding levels creates irregularly spaced contours.
4. Click the *Delete* button to remove the selected level from the list.
5. To modify an existing level, double-click the level value. Enter a new value for the contour level in the **Z Level** dialog. Click OK and the contour level list is updated.
6. When you are finished changing the contour levels, click OK or *Apply* and the map is displayed with your changes.



*The **Z Level** dialog is opened by double-clicking on a level number on the **Levels** page.*



Contour levels can be set to display irregular intervals between contours.

Level
20
25
30
35.742
40
45
50
55
60
65
70

Contour levels do not need to be regularly spaced.

Contour Lines

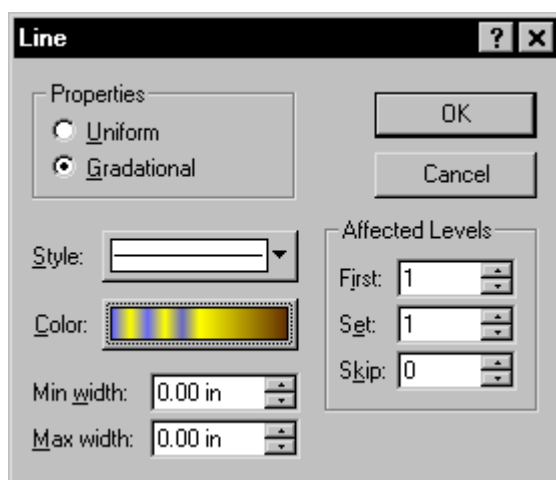
The contour line properties can be set using several methods. Uniform contour line properties are used to assign a single line style to a set of contour levels. Gradational contour line properties can be used to assign a series of gradational colors or line widths to the contour levels. For fine tuning, each individual contour line property can be set by double-clicking the line sample in the levels list.

Assigning Gradational Line Properties

Gradational line color and line widths can be assigned to contour lines.

To assign gradational line properties:

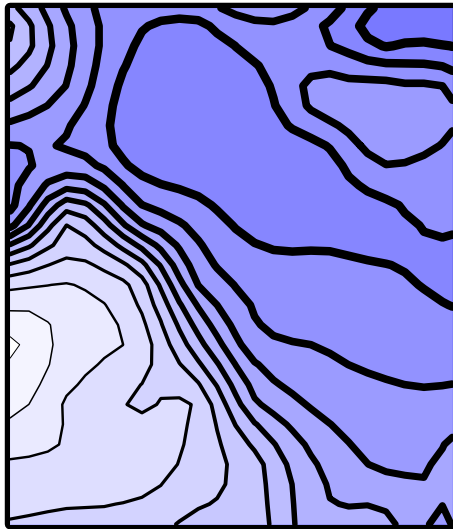
1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Click on the **Levels** page to edit the contour line properties.
3. On the **Levels** page, click the *Line* button to display the **Line** dialog.
4. In the **Line** dialog, select *Gradational*.
5. Click the line *Style* button to assign the same style to all of the contour lines.
6. Click the *Color* button to assign a color spectrum to the contour lines. The gradational color spectrum can be saved for use with other maps.



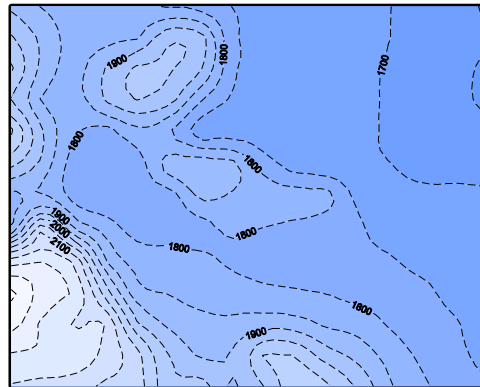
*Choose the gradational option in the **Line** dialog to vary the color and width of contour lines.*

Chapter 6 - Contour Maps

7. To assign varying line thickness, enter numbers into the *Min width* and *Max width* boxes. These line widths are applied to the minimum and maximum contour levels respectively. All intermediate line widths are interpolated proportionally.
8. If the *Skip* box is set to a number other than zero in the **Line** dialog, the contour lines are not all assigned the colors or widths set in the **Line** dialog.
9. Click OK in the **Line** dialog to return to the contour map properties dialog. The selected line colors and widths are shown on the **Levels** page. Be aware that some video cards and monitors are not capable of generating smooth gradations from one color to another, so the line spectrum might consist of ranges of one color, resulting in "color zones" rather than a smooth line color spectrum.
10. Click OK or *Apply* in the contour map properties dialog and the contour map is drawn with the properties set in the **Line** dialog.



This is a contour map using gradational line widths.



You can assign the same line properties to all contours on the map.

Assigning the Same Line Properties to all Contour Levels

To assign all contour levels the same line properties:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Click on the **Levels** page to edit the contour line properties.
3. On the **Levels** page, click the **Line** button to display the **Line** dialog.
4. In the **Line** dialog, choose *Uniform*.
5. Click the *Style* and *Color* buttons to open the line style and color palettes. Select the style and color to use for all of the contour lines. To eliminate the contour lines from the contour map, set the line style to *Invisible*.
6. Enter a number into the *Line width* box to set the line width. A line width of 0.00 is one pixel wide.
7. All of the levels are assigned the same line properties unless the number in the *Skip* box is set to a number other than zero.
8. Click OK to return to the contour map properties dialog.
9. Click OK or *Apply* in the contour map properties dialog to create a contour map with the same line properties for all contours on the map.

Setting Line Properties on a Frequency Basis

The *Affected Levels* settings are used to modify the contour levels on a frequency basis. These settings can be found by clicking the *Line* button on the **Levels** page in the contour map properties dialog. The *First*, *Set*, and *Skip* fields allow you to select which contour levels are modified.

- The *First* field indicates the first contour level affected by the change. This number refers to the contour level position in the list (starting with 1), not the actual contour value.
- The *Set* field indicates the number of levels to apply the changes.
- The *Skip* field indicates the number of contour levels to skip when assigning the properties.

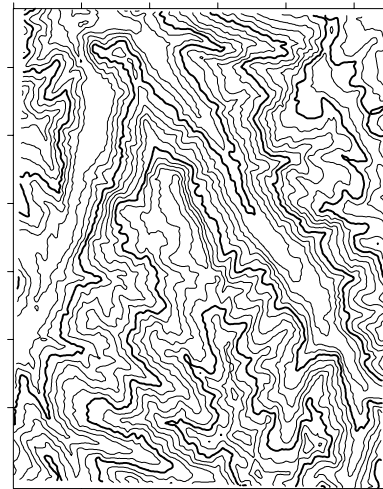
To use *Affected Levels* with lines:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Select the **Levels** page.
3. Click the *Line* button.
4. Set the first contour level to apply the properties to, the number of levels to be set, and the number of levels to skip.

Chapter 6 - Contour Maps

You can set the *Affected Levels* as many times as you want before leaving the dialog. For example, entering *First=one*, *Set=one*, *Skip=four* means apply the properties to the first contour level, skip four levels, set the next level, and so on. Then, you can enter *First=two*, *Set=four*, *Skip=one*. This applies the new properties starting at the second level. Four consecutive levels are modified, one is skipped, four are modified, one is skipped, etc. This combination of *Affected Levels* applies one set of properties to one group of levels and another set of properties to a second group of levels. Combinations of settings for line properties are limitless.

Levels	<i>First = 1</i> <i>Set = 1</i> <i>Skip = 4</i>	<i>First = 2</i> <i>Set = 4</i> <i>Skip = 1</i>
0	Yes	No
10	No	Yes
20	No	Yes
30	No	Yes
40	No	Yes
50	Yes	No
60	No	Yes
70	No	Yes
80	No	Yes
90	No	Yes
100	Yes	No



You can set the frequency of the line properties in the Affected Levels group.

Assigning Line Properties to Specific Contour Levels

To assign line properties to specific contour levels:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Click the **Levels** page.
3. Double-click the line sample for the contour line you want to modify.
4. The **Line Properties** dialog is displayed, allowing you to select the line style, color, and width.
5. Click OK in the **Line Properties** dialog and repeat the procedure for all the contour lines you want to change.
6. When you are done, click OK or *Apply* in the properties dialog and the map is drawn with the new line properties.

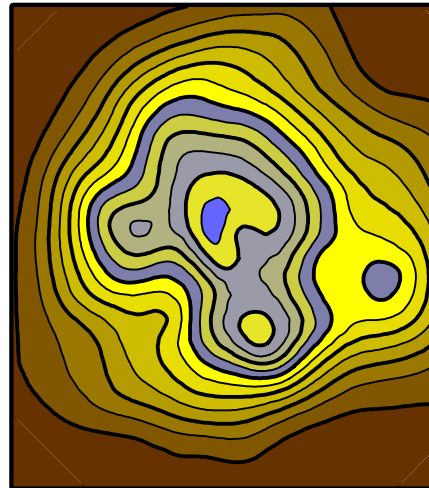
Index Contours

Index contours are lines of different colors, styles, or widths that appear at a regular frequency. You can create index contours by setting selected levels properties, or use the *Affected Levels* group in the **Line** dialog to automatically assign the index contours.

The *Affected Levels* group contains *First*, *Set*, and *Skip* fields. This group allows you to apply the selected line properties to a repeating series of levels. For example, use the *Affected Levels* options to create four blue contour lines, one red contour line, followed by four blue contour lines, etc. The *Affected Levels* group exists in the **Line**, **Fill**, **Labels**, and **Hachures** dialogs.

To create index contours:

1. Create a contour map.
2. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
3. Click the *Line* button on the **Levels** page.
4. Set the index contour line properties in the *Affected Levels* group in the **Line** dialog.
 - a. Click the line *Color* button and change the color to red.
 - b. In the *Affected Levels* group, set *First* to one, *Set* to one, and *Skip* to four. This means, starting with the first contour level, set one contour level to red, skip four contour levels, and then set the fifth contour level to red, skip four more levels, etc.
 - c. Click OK to enter the new level properties.
5. Next, set the line style and color used on the intermediate levels.
 - a. Click the *Line* button on the **Levels** page.
 - b. Click the line *Color* button and change the line color to blue.
 - c. In the *Affected Levels* group, set *First* to two, *Set* to four, and *Skip* to one. This means, starting with the second contour level (the first level is a red index contour) change the line color to blue for the next four contour levels, skip one contour level, set the next four contour levels to blue, etc.
6. Click OK in the **Line** dialog to return to the contour map properties dialog. The first contour level line appears as red, the next four are blue, followed by one red line, etc.
7. Click OK or *Apply* in the contour map properties dialog to draw the map with the index contours.

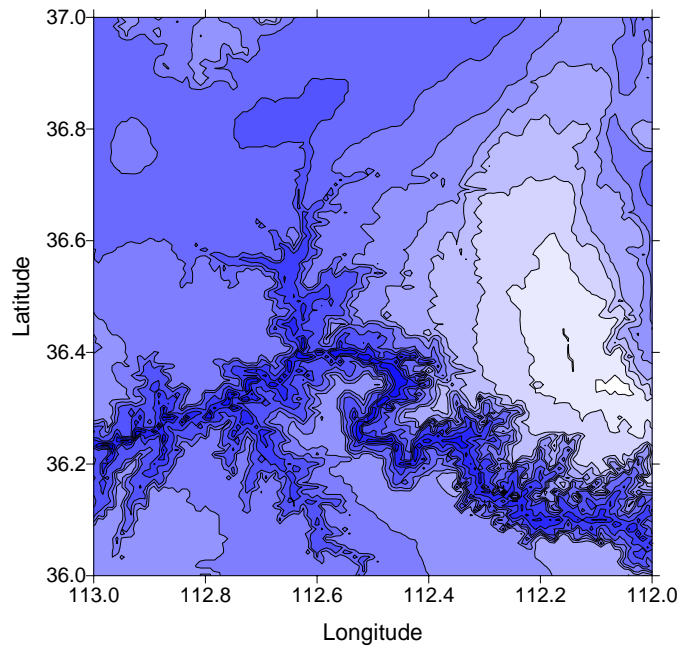


Create index contours through the *Affected Levels* group in the **Line** dialog.

Color Fill

Surfer allows you to fill the areas between contour lines with color fill. The color fill can gradationally change from the minimum to maximum contours, or specific fill properties can be assigned to individual contour levels.

Gradational colors are assigned in the **Color Spectrum** dialog. For example, if the color red is assigned to the minimum value, and the color blue to the maximum value, the resulting color spectrum graduates from red to blue. You can also assign fill properties to individual contour levels by double-clicking the fill sample on the **Levels** page in the contour map properties dialog. There is no gradational change for fill patterns.

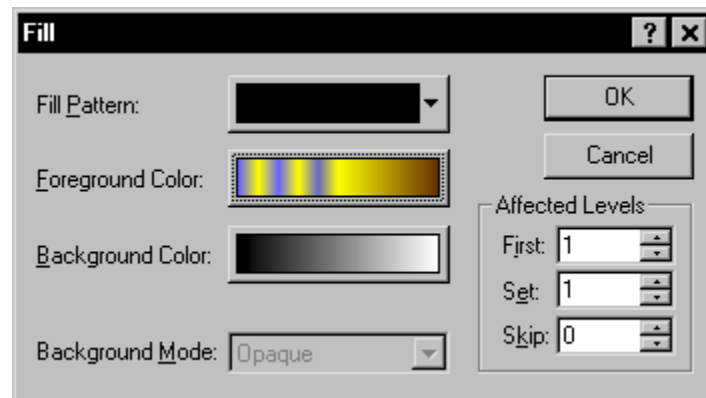


This is a filled contour map of a portion of the Grand Canyon.

Assigning Color Fill Based on a Fill Spectrum

Gradational fill color can be assigned to contour levels through the contour map properties dialog.

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section. Make sure the contour levels are set before proceeding with creating the color fill because adding or deleting a level after assigning a color spectrum alters the spectrum.
2. Click the *Fill* button on the **Levels** page to display the **Fill** dialog.
3. Click the *Fill Pattern* button to display the pattern palette. The fill pattern is constant for all levels.
4. Click the *Foreground Color* or *Background Color* buttons to open the **Color Spectrum** dialog. The foreground color is used for the pattern. The background color is used for areas behind the pattern. The *Background Mode* can be set to *Opaque* or *Transparent* with Windows stock fills and vector fills. With bitmap fills, the background is always set to *Opaque*.
5. Click OK in the **Fill** dialog to return to the contour map properties dialog. The new fill properties are displayed on the **Levels** page.
6. Check the *Fill Contours* option on the **General** page.
7. Click OK or *Apply* and the contour map is drawn with colors filling the areas between the contour lines.



*The **Fill** dialog is opened by clicking the Fill button on the **Levels** page in the contour map properties dialog.*

Assigning Color Fill to Specific Contour Levels

To assign fill properties to specific contour levels:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Click the **Levels** page.
3. Double-click the fill sample for the contour level you want to modify.
4. Select the fill properties from the **Fill Properties** dialog.
5. Click OK in the **Fill Properties** dialog and repeat the procedure for all the contour levels you want to change.
6. Check the *Fill Contours* option on the **General** page.
7. Click OK or *Apply* and the contour map is drawn with colors filling the areas between the contour lines.

Setting Fill Properties on a Frequency Basis

The *Affected Levels* settings are used to modify the contour levels on a frequency basis. These settings can be found by clicking the *Fill* button on the **Levels** page in the contour map properties dialog. The *First*, *Set*, and *Skip* fields allow you to specify which contour levels are modified.

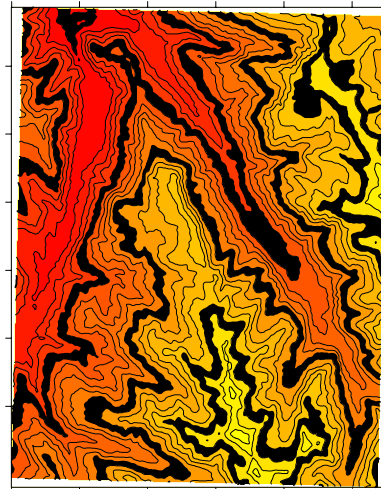
- The *First* field indicates the first contour level affected by the change. The number refers to the contour level position in the list (starting with 1) and not the actual contour value.
- The *Set* field indicates the number of levels to apply the changes.
- The *Skip* field indicates the number of contour levels to skip when assigning the specified properties.

To use *Affected Levels* with fills:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Select the **Levels** page.
3. Click the *Fill* button.
4. Set the first contour level to apply the properties to, the number of levels to be set, and the number of levels to skip.
5. Set the fill properties.
6. Click OK.

You can set the *Affected Levels* as many times as you want before leaving the dialog. For example, entering *First=one*, *Set=one*, *Skip=four* means apply the properties to the first contour level, skip four levels, set the next level, and so on. Then, you can enter *First=two*, *Set=four*, *Skip=one*. This applies the new properties starting at the second level. Four consecutive levels are modified, one is skipped, four are modified, one is skipped, etc. This combination of *Affected Levels* applies one set of properties to one group of levels and another set of properties to a second group of levels. Combinations of settings for fill properties are limitless.

Levels	<i>First = 1</i> <i>Set = 1</i> <i>Skip = 4</i>	<i>First = 2</i> <i>Set = 4</i> <i>Skip = 1</i>
0	Yes	No
10	No	Yes
20	No	Yes
30	No	Yes
40	No	Yes
50	Yes	No
60	No	Yes
70	No	Yes
80	No	Yes
90	No	Yes
100	Yes	No



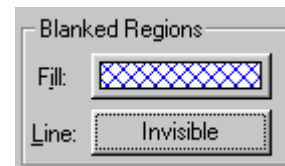
You can set the frequency of the fill properties in the Affected Levels group.

Creating a Filled Contour Map Containing Blanked Areas

When a filled contour map contains blanked areas, the blanked area has a default color assigned to it. You can select the fill for the blanked areas as well as the line surrounding the blanked areas on the **General** page in the contour map properties dialog. Blanked areas are typically created with the **Grid | Blank** command. Blanked areas can also be created due to insufficient data or missing during the gridding process.

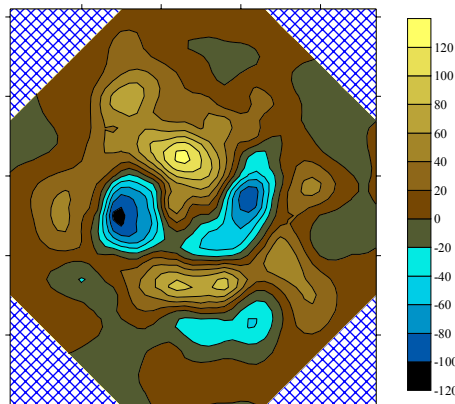
To assign fill and line properties to blanked areas:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Click the **General** page.
3. In the *Blanked Regions* group, click the **Fill** button to open the **Fill Properties** dialog. Choose the *Fill Pattern*, *Foreground color*, *Background color*, and *Scale Factor* (vector patterns only) for the blanked areas.
4. Click OK in the **Fill Properties** dialog to return to the contour map properties dialog.
5. In the *Blanked Regions* group, click the **Line** button to open the **Line Properties** dialog. Choose the line *Style*, *Color*, and *Width* for the line surrounding blanked areas.
6. Click OK in the **Line Properties** dialog to return to the contour map properties dialog.
7. Click OK or *Apply* in the contour map properties dialog to create the map with the selected line and fill for the blanked areas.



*You can set the line and fill properties for the blanked areas on the **General** page in the contour map properties dialog.*

The fill and line properties for the blanked areas do not appear in the color scale bar.



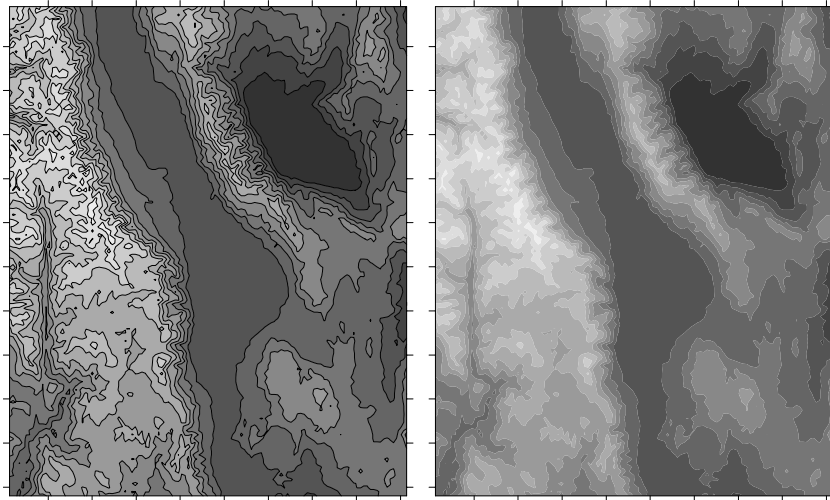
*You can set the line and fill properties for blanked areas on the **General** page in the contour map properties dialog. This map contains blanked areas in the four corners of the map. The blanked areas have a diagonal cross pattern. The blanked fill pattern does not appear on the color scale bar.*

Displaying a Filled Contour Map without Contour Lines

To create a filled contour map without contour lines set the contour line properties to *Invisible*. In the **Line** dialog, the invisible line style is represented by the blank box in the style palette.

To eliminate lines from a contour map:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Click the **Levels** page.
3. Click the *Line* button.
4. In the **Line** dialog, click the line *Style* button and choose the *Invisible* line style from the list.
5. Click OK in the **Line** dialog to return to the contour map properties dialog.
6. Click OK or *Apply* in the contour map properties dialog, and the filled contour map is displayed without contour lines.



This is the same filled contour map displayed with and without contour lines.

Color Scale Bar

A color scale bar is a legend that shows the fill assigned to each contour level on a filled contour map.

Adding a Color Scale Bar

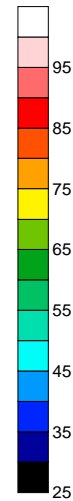
To add a color scale bar to a filled contour map:

1. In the contour map properties dialog, specify the fill properties to use for the contour levels.
2. Check the *Fill Contours* and *Color Scale* options on the **General** page to activate the color scale bar.
3. Click OK or *Apply* to draw the contour map with a color scale bar.

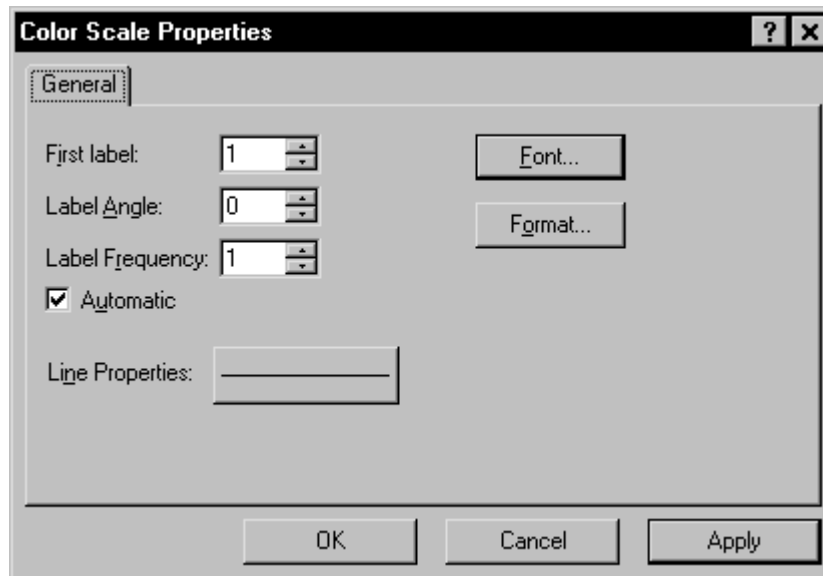
Editing a Color Scale Bar

To modify a color scale bar:

1. Double-click the scale bar to display the **Color Scale Properties** dialog.
 - Indicate the level index number of the first label in the *First* label box. The level index number represents the relative position of the contour level in the ordered list (starting at one), and not the actual level value.
 - Choose the angle at which the labels are drawn in the *Label Angle* box.
 - Use the *Label Frequency* field to specify how often a label appears. A frequency of one draws every label, two draws every other label, etc.
 - The *Automatic* check box sets the label frequency so labels do not overlap. If there are too many labels to fit in the space next to the color scale bar, some labels are dropped to make the labels legible.
 - Click the *Font* button to set the label text properties.
 - Click the *Format* button to choose the label numeric format (i.e. number of decimal places).
 - The *Line Properties* button allows you to set the line properties for the color scale bar border.
2. Once the changes are made, click OK or *Apply* to redraw the color scale bar.



Color scale bars are added through the contour map properties dialog.



*The **Color Scale Properties** dialog controls the display of the color scale bar.*

Color Scale Bar Visibility

The color scale bar can be toggled on and off through the **Object Manager**.

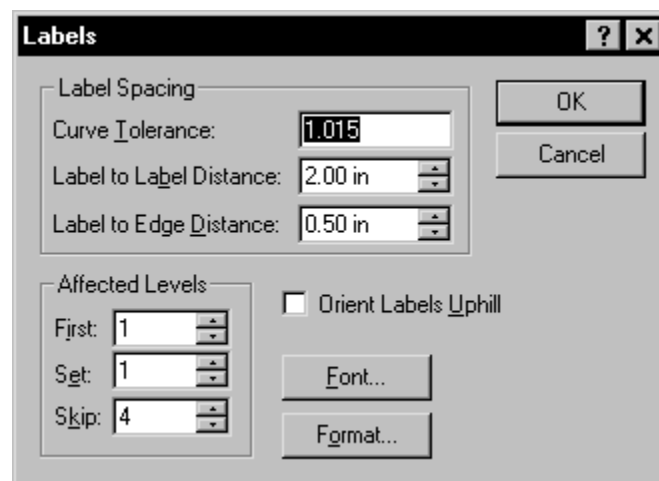
Removing a Color Scale Bar

A color scale bar can be removed by clicking on the color scale bar and pressing the DELETE key on your keyboard. Alternatively, you can uncheck the *Color Scale* check box in the map properties dialog.

Labels

Contour labels indicate the value of the contour line. Labels are regularly spaced along the contour line subject to curvature restrictions. Contour labels can use any text properties and numeric format, but all contour labels in a given map must use the same properties.

Contour labels are controlled from the contour map properties dialog. In the *Label* column on the **Levels** page, the labeled contour levels are indicated by the word *Yes*. To change the display of contour labels for a particular level, double-click the words *Yes* or *No* in the list.



*The **Labels** dialog is opened by clicking on the Label button on the **Levels** page in the contour map properties dialog.*

Displaying Contour Labels at a Regular Interval

To display contour labels:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Click the **Levels** page.
3. Click the *Label* button to display the **Labels** dialog. Use the *First*, *Set*, and *Skip* boxes to control the labeled level frequency. These values correspond to the relative numeric position (starting at one) in the *Levels* list, and not the actual Z values.
 - The *Label Spacing* group controls the placement of contour labels along the individual contour lines.
 - The *Affected Levels* group controls which levels are labeled.

- The *Orient Labels Uphill* check box displays the labels so they are always oriented uphill. If this box is checked, the "tops" of the labels point uphill. If this option is unchecked, the labels always are oriented right side up on the page.
 - The *Font* button displays the **Text Properties** dialog. Choose the face, point size, style, and color to use for the contour labels in this dialog.
 - The *Format* button displays the **Label Format** dialog. This dialog allows you to specify the numeric format to use for the contour labels (i.e. number of decimal places).
4. Click OK and the *Label* list on the **Levels** page indicates the labeled contour levels.
 5. Click OK or *Apply* and the map is drawn with labels on the specified contour levels.

Assigning Labels to Specific Contour Levels

Labels can be assigned to any contour level by double-clicking on the words *No* and *Yes* in *Label* list on the **Levels** page.

Setting Label Properties on a Frequency Basis

The *Affected Levels* settings are used to modify the contour levels on a frequency basis. These settings can be found by clicking the *Label* button on the **Levels** page in the contour map properties dialog. The *First*, *Set*, and *Skip* fields allow you to specify which contour levels are modified.

- The *First* field indicates the first contour level affected by the change. The number refers to the contour level position in the list (starting with 1) and not the actual contour value.
- The *Set* field indicates the number of levels to apply the changes.
- The *Skip* field indicates the number of contour levels to skip when assigning the specified properties.

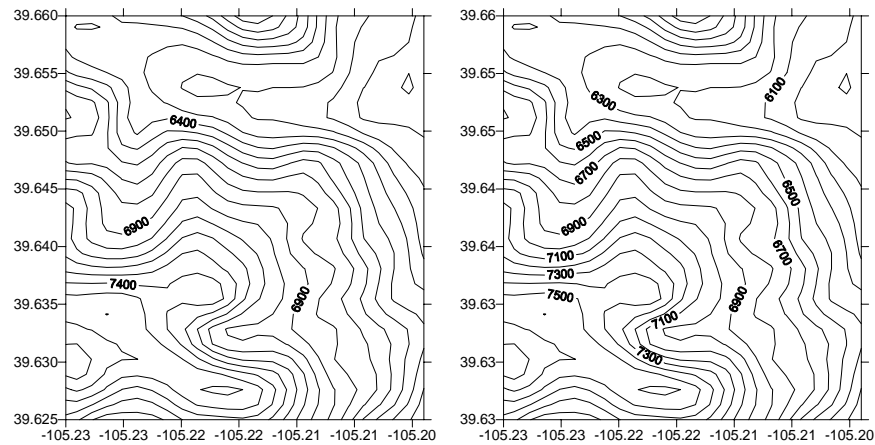
To use *Affected Levels* with labels:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Select the **Levels** page.
3. Click on the *Label* button.
4. Set the *First* contour level affected by the setting, the *Set* number indicating the number of levels to set with the specified properties, and the *Skip* number indicating the number of levels to skip assigning the specified properties.

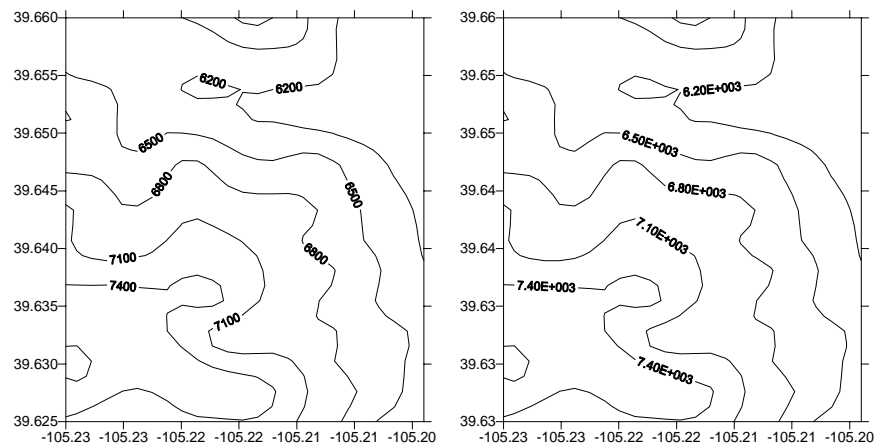
Levels	First = 1 Set = 1 Skip = 4
6000	Yes
6100	No
6200	No
6300	No
6400	No
6500	Yes
6600	No
6700	No
6800	No
6900	No
7000	Yes

Chapter 6 - Contour Maps

Labels can only have one set of *Affected Levels*. If the *Affected Levels* are set a second time, they override the changes made by the first.



Contour labels can be placed on contour lines at any interval. In these examples, Skip is set to four for the map on the left, and one for the map on the right.



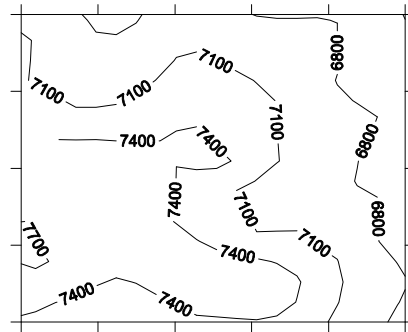
Contour labels can use different numeric formats. The map on the left uses the fixed format with zero decimal digits, and the map on the right uses the exponential format with two decimal digits.

Label Spacing

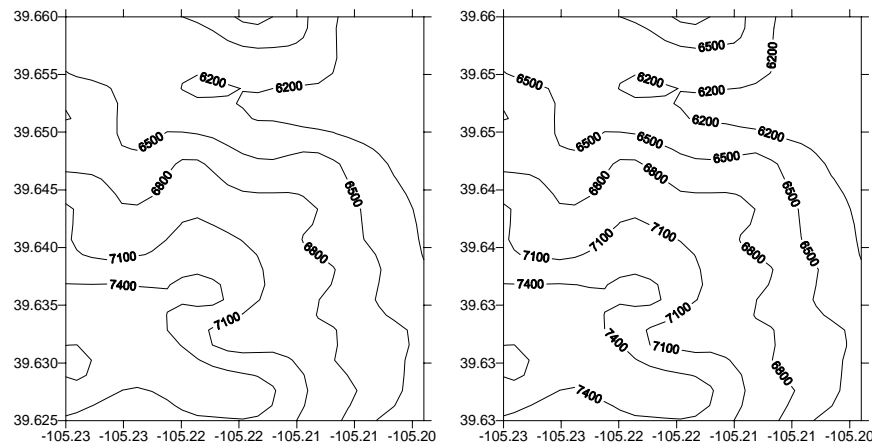
Contour labels are automatically placed on a straight section of the contour line. Three parameters control the spacing of the labels: *Curve Tolerance*, *Label to Label Distance*, and *Label to Edge Distance*. These parameters are set in the **Labels** dialog accessed through the **Levels** page in the contour map properties dialog.

Curve Tolerance

Curve Tolerance specifies the maximum amount of contour curvature allowed when placing labels on contour lines. Curve tolerance is calculated by dividing the actual distance along the contour line by the straight-line distance between the end points of the contour label. Highly curved lines might not be labeled automatically. You can increase the curve tolerance value to allow labels on highly curved contour lines, although contour labels might be hard to read. The default *Curve Tolerance* value of 1.015 should be acceptable in most cases.



Increasing the Curve Tolerance allows labels to be drawn on highly curved contours.



Using a small Label to Label Distance and a small Label to Edge Distance can greatly increase the number of contour labels on a map.

Label to Label Distance

Label to Label Distance specifies the minimum distance (in inches or centimeters) between labels along the contour line. **Surfer** searches for the next suitable location for a label after moving the minimum distance specified by the *Label to Label Distance*. The next label is not drawn until a

Chapter 6 - Contour Maps

segment of the necessary length, within the curve tolerance limits, is found. As the *Label to Label Distance* is increased, fewer labels are drawn on the contours.

Label to Edge Distance

Label to Edge Distance specifies the minimum distance (in inches or centimeters) from the label to the edge of the map. This feature controls label placement so labels do not overwrite the map borders or axes.

Removing All Labels

To avoid displaying contour labels on the map:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Click the **Levels** page.
3. Click the *Label* button.
4. In the **Labels** dialog, change the *Set* value to zero.
5. Click OK in the **Labels** dialog to return to the contour map properties dialog.
6. Click OK or *Apply* in the contour map properties dialog to remove all labels.

Editing Label Positions

Contour labels can be interactively moved, added, and deleted.

Entering Edit Mode

To enter contour label edit mode,

1. Select a contour map by clicking on it in the plot window or by clicking on it in the Object Manager.
2. Select the **Map | Contour Map | Edit Labels** command.

Alternatively, you can right-click on the selected map and choose **Edit Contour Labels** from the pop-up menu. When this command is activated, each contour label is displayed with a rectangular outline.

Moving Labels

To move a label, select it with the mouse or keyboard and drag it to a new location. To drag with the mouse, hold down the left mouse button and move the cursor to a new location. To drag with

the keyboard, position the cursor on top of the label, hold down the SPACEBAR, and press the ARROW keys.

Moving Off Screen

If you are zoomed in on the contours, use the scroll bars to move to off window locations.

Canceling Movement

To cancel label movement and to return the label to its original position, press the ESC key before releasing the left mouse button or SPACEBAR.

Deleting Labels

To delete a label, select it and press the DELETE key.

Adding New Labels

To add a contour label, hold down the CTRL key and click with the mouse on a contour. A label is not added if the contour is too small or too curved for a label to fit.

Undo

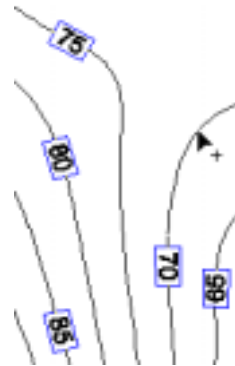
Click on **Edit | Undo** to undo a label position, deleting a label, or adding a label.

Ending Edit Mode

To leave contour label edit mode, press the ESC key or choose the **Edit Labels** command again.

Losing Label Edits

If you make any changes to the contour map through the contour map properties dialog, changes to the contour labels are deleted. **Surfer** displays a warning message that the custom labels will be lost in this case.



*When **Map | Contour Map | Edit Labels** is selected, each contour label is surrounded by a blue box. Add a contour label by holding down the control key and clicking on the location to add the label. The plus symbol next to the cursor indicates add label mode.*

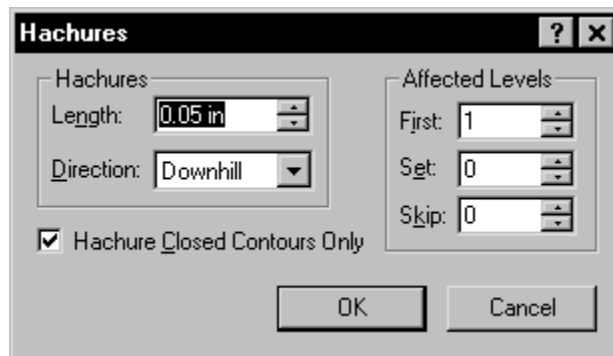
Hachures

Hachures are small tick marks placed along contour lines to indicate the direction of slope. In **Surfer**, hachures can point either up slope or down slope. Hachure display is controlled from within the **Levels** page in the contour map properties dialog.

Setting Hachures on a Regular Interval

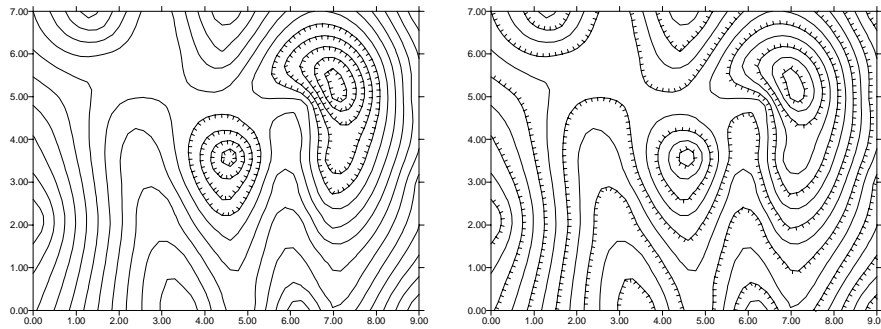
To place hachures on a contour map at a regular interval:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. To display hachures on regular intervals, click the *Hach* button to open the **Hachures** dialog.
 - Use the *Length* box to control the length of the hachures. Hachure length can be greater than zero, and up to one inch.
 - The *Direction* box controls whether hachures point in an upward or downward direction. For example, to point hachures towards contours of lower value, choose the *Downhill* option.
 - The *Hachure Closed Contours Only* check box ensures that only closed contour lines use hachures. If a contour line intersects the border of the map, hachures are not displayed on that contour line. If the *Direction* is set to *Uphill*, only peaks are hachured. If the *Direction* is set to *Downhill*, only depressions are hachured.
 - The *Affected Levels* group controls which levels are hachured. For more information on the use of *First*, *Set*, and *Skip* see *Setting Hachures on a Frequency Basis*.
3. Make the desired changes and click OK to return to the **Levels** page in the contour map properties dialog. The *Hach* list is updated to indicate the hachured contour levels. Click OK or *Apply* and the map is drawn with hachures on the specified contour levels.



Open the **Hachures** dialog by clicking the *Hach* button on the **Levels** page in the contour map properties dialog.

As an example, consider a map displaying contour lines every ten feet from 50 to 100. Within the **Hachures** dialog, if you specify 1 (corresponding to the first contour line in the levels list) as the *First* hachured contour line, and a *Skip* value of one, then hachures appear on the contour lines at 50, 70, and 90 (every second contour line on the map).



These are two examples showing different hachuring parameters. The map on the left displays hachures on the uphill side of contours on closed contours only. The map on the right displays hachures on both open and closed contours, and the hachures are placed on the downhill side of the contours.

Setting Hachures on Specific Levels

In the *Hach* column on the **Levels** page, the hachured contour levels are indicated by the word *Yes*. Contours not hachured are indicated with the word *No*. To control the display of hachures for a particular level, double-click the words *Yes* or *No* in the list.

Setting Hachures on a Frequency Basis

The *Affected Levels* settings are used to modify the contour levels on a frequency basis. These settings can be found by clicking the *Hach* button on the **Levels** page in the contour map properties dialog. The *First*, *Set*, and *Skip* fields allow you to specify which contour levels are modified.

- The *First* field indicates the first contour level affected by the change. The number refers to the contour level position in the list (starting with 1) and not the actual contour value.
- The *Set* field indicates the number of levels to apply the changes.
- The *Skip* field indicates the number of contour levels to skip when assigning the specified properties.

Chapter 6 - Contour Maps

To use *Affected Levels* with hachures:

1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Select the **Levels** page.
3. Click on the *Hach* button.
4. Set the *First* contour level affected by the setting, the *Set* number indicating the number of levels to set with the specified properties, and the *Skip* number indicating the number of levels to skip assigning the specified properties.

Hachures can only have one set of *Affected Levels*. If the *Affected Levels* are set a second time, they override the changes made by the first.

Removing All Hachures


To avoid displaying hachures on the map:

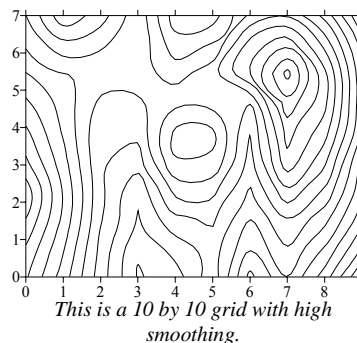
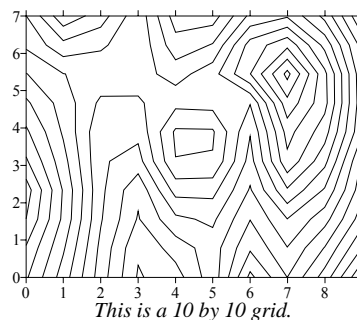
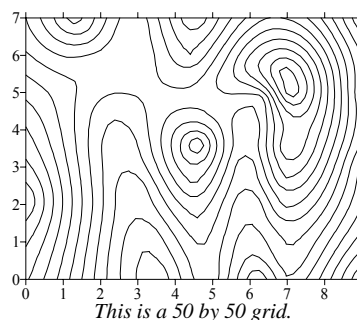
1. Open the map properties dialog using one of the methods listed in the *Editing Contour Map Properties* section.
2. Click the **Levels** page.
3. Click the *Hach* button.
4. In the **Hachures** dialog, change the *Set* value to zero.
5. Click OK in the **Hachures** dialog to return to the contour map properties dialog.
6. Click OK or *Apply* in the contour map properties dialog to remove all labels.

Smoothing Contours

Contour smoothness controls the roundness of the contour lines. Smoothness in a contour map can be controlled in several ways:

1. Produce a denser grid file from the original data. As a rule, contour maps made from high-density grids are visibly smoother than contour maps made from low-density grid files. For example, a 10 X 10 grid file (10 rows and 10 columns) results in more angular contours than a 50 X 50 grid file.
2. Using the **Grid | Spline Smooth** command to insert additional rows and columns into the grid file.
3. Check the *Smooth Contours* option in the contour map properties dialog. If smoothing is enabled in the contour map properties dialog, contour lines may cross. The *Smooth Contours* option in the contour map properties dialog applies a constrained spline algorithm to interpolate additional contour vertices.
4. Change the gridding method or gridding method parameters. Some methods produce smoother grids than others, and some have smoothing parameters that can be utilized during gridding. See *Chapter 4, Creating Grid Files* for more information on gridding.

If you create a new grid file, click the  button to replace the old grid in the contour map.



Export Contours

To retain the Z information for contour lines for use in other mapping programs, use the **Map | Contour Map | Export Contours** command. **Export Contours** exports the selected contour map as an ASCII AutoCAD DXF file. The contours are saved as elevated polylines. The polylines are clipped to the map limits and scaled to the current map scale. No axes, labels, contour fills, line properties (width, style, and color), etc. are exported. The contours are not affected by 2D page transformations or rotations. The coordinates in the AutoCAD DXF file are saved with 64-bit floating-point precision.

To export 3D contour lines:

1. Create a contour map.
2. Select the contour map by clicking on the map name in the **Object Manger**. It does not matter if the contour map is overlaid on other maps as long as the contour map is selected.
3. Click on **Map | Contour Map | Export Contours** to display the **Save As** dialog. Enter the name of the file and click *Save*. The file is exported with a [.DXF] extension.

If you wish to export the contours as part of a complete map, choose one of the export options in **File | Export**.

Chapter 7

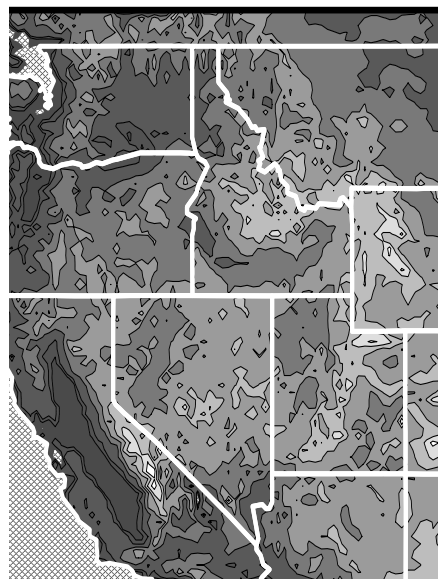
Base Maps

Introduction to Base Maps

Base maps display geographic and political information such as roads, streams, lakes, or state and county boundaries. Base maps can be combined with any other map type in **Surfer**.

The **Map | Base Map** command creates base maps from existing boundary files. You can use the base map dialog to set the base map line, fill, symbol, and font properties. With bitmaps you can also reassign the base map coordinates.

Boundary maps show objects at precise XY locations on a map. Boundary maps contain point locations, polygons, polylines, text, and bitmaps. The base map limits for boundary maps are defined by the XY limits of the objects in the imported boundary file.



This is a contour map of the western United States with a base map overlay of state outlines.


Base Map File Types

Base map file types currently supported by **Surfer** include:

- Golden Software Boundary [.GSB]
- Atlas Boundary [.BNA]
- USGS Digital Line Graph [.DLG], [.LGO], [.LGS]
- AutoCAD DXF Drawing [.DXF]
- Golden Software PlotCall [.PLT]
- Golden Software Blanking [.BLN]
- Windows Clipboard (Picture) [.CLP]
- Windows Metafile (Picture) [.WMF]
- Enhanced Metafile [.EMF]
- ESRI Shapefile [.SHP]
- ESRI ArcInfo Export Format [.E00]
- MapInfo Interchange Format [.MIF]
- Windows Bitmap [.BMP]
- JPEG Compressed Bitmap [.JPG]
- Portable Network Graphics Bitmap [.PNG]
- Tagged Image [.TIF]
- Targa (TrueVision) [.TGA]
- PCX (ZSoft/Paintbrush) [.PCX]
- DCX (Multi-page PCX) [.DCX]
- CompuServe Bitmap [.GIF]
- WordPerfect Bitmap [.WPG]
- PICT (MacIntosh) Bitmap [.PCT]
- SDTS Topological Vector Profile [.DDF]
- Golden Software Interchange [.GSI]

Creating a Base Map

To load a base map file:

1. Choose the **Map | Base Map** command or click the  button.
2. The **Import File** dialog is displayed with a list of files in all the format types that can be imported. Select a boundary file in the list and click *Open*.
3. An import options dialog for the file type is displayed. Make any changes to the format-specific import options if necessary.
4. Choose OK and the base map is imported into the current plot window.

Editing Base Map Properties

To change the features of a base map, open the base map properties dialog by:

- double-clicking on the map in the plot window,
- double-clicking on the map name in the **Object Manager**,
- right-clicking the map and choosing Properties in the context menu,
- or by clicking **Edit | Properties** when the map is selected.

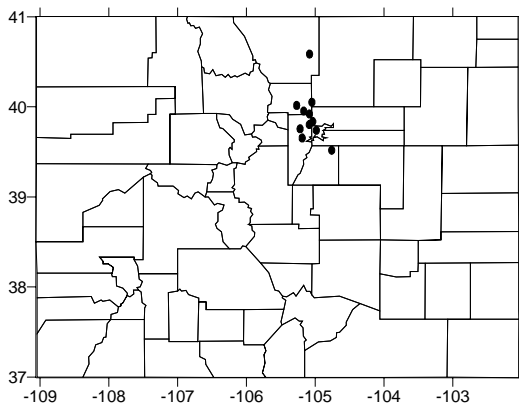
The features that can be edited on a base map vary depending on the base map file type. Generally, you can edit the coordinates of bitmaps and the lines, symbols, fill, and text properties in a vector file.

Base Map Properties

Line and fill properties can be applied to all the polygons on the map, but all polygons must use the same properties. For example, choosing a red fill for the polygons on the base map causes all polygons on the selected map to be red. You cannot select individual polygons on the map and assign unique fill or line properties to them. For polylines, assigned line properties are the same for all polylines. This is the same as the line properties assigned to the lines defining the polygons on the base map. If the base map displays point locations (represented by symbols), the same symbol properties are assigned to all the point locations.

Boundaries that comprise the objects in a base map cannot be modified. For example, if a base map contains a polygon, you cannot select the polygon and change the shape of the polygon. Similarly, individual point symbols on a base map cannot be moved or deleted.

The base map properties dialog contains five pages: **Base Map**, **View**, **Scale**, **Limits**, and **Background**. **View**, **Scale**, **Limits**, and **Background** are discussed in *Chapter 14, Common Map Properties*.



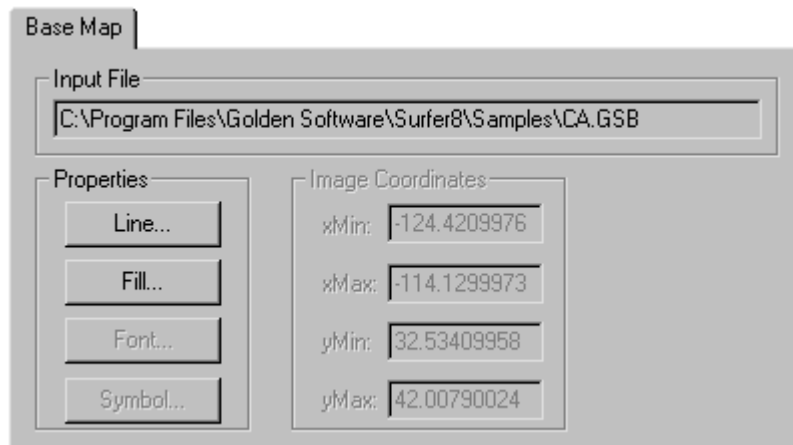
A base map can contain polygons, polylines, and points. In this map, the polygons are state and county boundaries and the points are cities.

Base Map

The **Base Map** page in the base map properties dialog contains options for setting vector map properties and bitmap coordinates.

The base map properties **Base Map** page contains the following options:

- The *Input File* lists the current file used in the base map.
- The buttons in the *Properties* group can be used to set vector file line, fill, font, and symbol properties. If a button is grayed out, there are no objects of that type in the base map. For example, some programs export polygons as polylines. If this is the case, the *Fill* button is grayed out.
- The *Image Coordinates* group provides a way to set the corner coordinates for bitmaps.



Since the Line and Fill buttons are activated, the base map is a vector map containing lines and polygons.

Changing Line, Fill, Font, or Symbol Properties in a Base Map

Properties can be assigned to base map polylines, polygons, text, or symbols if the base map consists of a vector file. Any assigned properties are applied to all the objects of similar type in the base map. See *Appendix C, File Formats* for more information on file formats.

To change the properties of objects in a base map:

1. Open the map properties dialog using one of the methods listed in the *Editing Base Map Properties* section.
2. In the **Base Map** page, click the button for the type of property you want to change. If the button is grayed out, the base map does not contain any objects of that type. For example, if the base map does not contain any polygons, the *Fill* button is grayed out.
3. Specify the properties you want to assign to the objects in the base map.
4. Click OK or *Apply* in the property dialog (**Line Properties**, **Fill Properties**, **Text Properties**, or **Symbol Properties**).
5. Click OK in the base map properties dialog, and the base map is redrawn. The new properties are applied to all the polyline, polygon, text, or symbols in the base map.



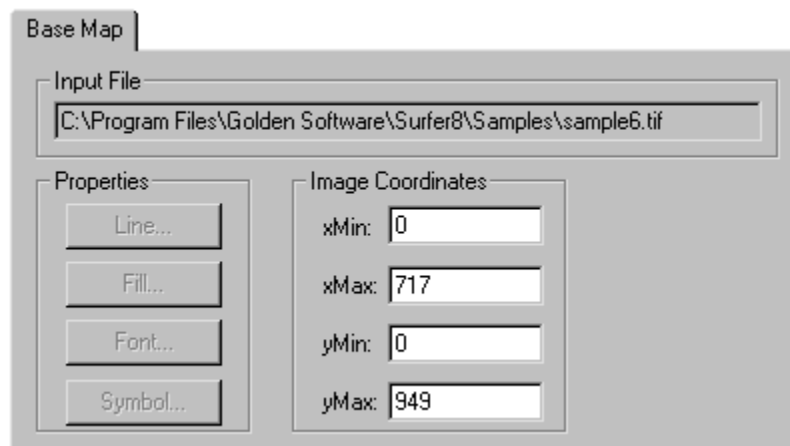
This example shows two overlaid base maps. One base map has thicker lines, assigned through the base map properties dialog.

Assigning Coordinates to a Bitmap File

You can redefine bitmap coordinates imported using the **Map | Base Map** command (not **File | Import**). By default, bitmap files are assigned a coordinate system that corresponds to the rows and columns in the bitmap. Under most circumstances, these coordinates do not match the coordinate systems used on other types of maps. In this case, redefine the coordinates for the bitmap. After the bitmap coordinates are redefined, the bitmap base maps can be used as overlays with other types of maps. The image coordinate options are grayed unless the base map consists of a single bitmap.

To change the bitmap coordinates:

1. Use the **Map | Base Map** command to import the bitmap.
2. Open the map properties dialog using one of the methods listed in the *Editing Base Map Properties* section.
3. In the *Image Coordinates* group, type the *xMin*, *xMax*, *yMin*, and *yMax* values to apply to the bitmap image.



When a bitmap is used as a base map, you can assign new coordinates to the file through the base map properties dialog.

4. Click OK or *Apply* and the bitmap is redrawn using the new coordinates.

Placing Boundaries on Other Maps

Base maps can be used to place boundary features on any type of map. When you want to place boundary lines on a map, create the base map, select all maps, and then use the **Map | Overlay Maps** command to combine the maps. The maps must use the same coordinate system or they will not overlay correctly.


Coordinates

Base map file coordinate systems must use the same units and be reasonably close to each other to avoid the extreme scaling of one map or the other. **Surfer** merges the coordinate systems of the individual maps. For example, consider a contour map based on latitude/longitude coordinates. To overlay a base map on the contour map, the base map must also use latitude/longitude coordinates in the same general range as the contour map.

Limits and Scale

When overlaying maps, the composite map limits are based on the greatest extent determined from all the overlays included in the composite map. For example, you may have a base map that covers a much greater extent than the contour map on which you want to display boundary information. When you overlay these maps, the limits are based on the overlay with greater extent, and the other overlays might be quite small within the limits of the composite map. When this occurs, you can clip the limits of the map to the smaller overlay limits using the map properties **Limits** page. The scale can then be changed using the map properties **Scale** page.


To place a boundary on a map:

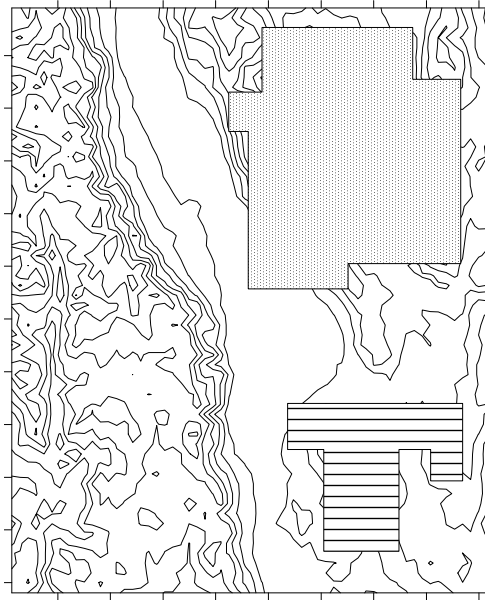
1. Create a map, such as a contour or wireframe. The grid file should use limits that cover the same general XY extent as the base map.
2. Choose the **Map | Base Map** command, or click the  button.
3. Select the file containing the boundaries you want to include on the map. Click *Open* and an import options dialog appears.
4. Select import options for the map and click OK. The base map is drawn.
5. Select the maps. If there are the only maps in the plot window, click **Edit | Select All** to select both maps.
6. Choose the **Map | Overlay Maps** command to create a single composite map.
7. To clip the map to specific limits, open the map properties and select the **Limits** page. If necessary, set the map scale on the **Scale** page.
8. Click OK or *Apply* to redraw the map.

Masking Portions of a Map

To mask areas of a map, you can overlay a base map or filled shape on the map. The base map must contain polygons that can be filled with patterns or colors that obscure the underlying map.

To overlay a masking base map:

1. Create a map of your grid file or data file.
2. Choose the **Map | Base Map** command and select a vector base map. The base map must contain polygons (areas), so do not check the *Areas to Curves* option during base map import.
3. Click **Edit | Select All**.
4. Select the **Map | Overlay Maps** command. The map and base map are combined into a single composite map.
5. Right-click on the base map in the **Object Manager** and then select **Order Overlay | Move to Front** from the pop-up menu. The base map moves to the top of the list. (Open the **Object Manager** with **View | Object Manager** or by clicking the  button.)
6. While the base map is still selected, click the **Edit | Properties** command.
7. Click the *Fill* button on the **Base Map** page and specify the fill properties to apply to the base map.
8. Click OK or *Apply* to redraw the boundaries that mask the underlying map.



This map contains two base maps on top of a contour map. A fill pattern is added to the boundaries on the base maps. The boundaries obscure the underlying contours.

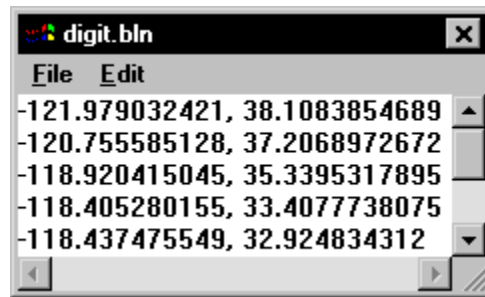
If exact masking boundaries are not required, you can use one of the drawing tools such as the polygon, rectangle, or ellipse tools to draw objects on top of a map. If fill is added to these types of objects, you can mask portions of a map without creating a boundary file. Be aware that drawing objects cannot be overlaid on map objects. This means that resizing or scaling the map does not resize the drawing object with it.

Digitize

The **Map | Digitize** command activates the display of map coordinates on the status bar, and allows you to write map coordinates to a data file. As you move the pointer across the selected map, the XY map coordinates for the current mouse position are shown in the status bar. Click the left mouse button, and the digitize editor window is displayed. The coordinates for the clicked point are written to the editor window. Each time the map is clicked, a small, temporary red symbol is drawn on the map, and the map coordinates for the current mouse position are written to the digitize editor window. In this way, you can digitize boundaries from maps and easily create boundary files from the digitized information. To use the **Digitize** command, you must select a horizontal (Tilt = 90) planar map.

To use the **Digitize** command:

1. Create a map in the plot window and select the map. You can select an overlay, although the overlay cannot contain a wireframe or surface, or be tilted at an angle other than 90 degrees.
2. Choose the **Map | Digitize** command and the cursor becomes a cross hair cursor.
3. As you move the cross hair cursor within the plot window, the map coordinates for the position are displayed in the status bar.
4. Click the left mouse button in the plot window to write the current coordinates to the digitize editor window. Continue adding points in this manner.
5. When you are finished using the **Digitize** command, click on any button or command, or press the ESC key. You are prompted to save the data as a Golden Software Blanking file [.BLN] or a Golden Software Data file [.DAT]. Select *Yes* to save the data, *No* to discard the data, or *Cancel* to return to the map and remain in the digitize mode.



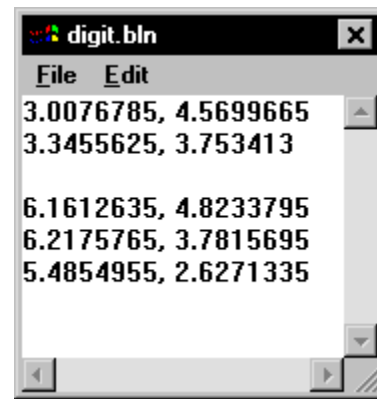
*The digitize editor window displays the map coordinates as you click on the map. You can save the coordinates to a file using the **File | Save** command.*

Creating a Blanking File with the Digitize Command

One of the uses of the **Map | Digitize** command is to create a Golden Software Blanking file [.BLN]. When creating a blanking file in this manner the header information is added to the file automatically. The file automatically blanks inside the polygon. If you wish to blank outside the polygon, edit the file in an ASCII editor or the **Surfer** worksheet to change the blanking value from 1 (blank inside) to 0 (blank outside). See *Appendix C, File Formats* for details on the blanking file format.

To create a blanking file with the **Digitize** command:

1. Create a map in the plot window, and then select it. This can be a composite map, although the composite map cannot contain a wireframe or surface, or be tilted at any angle other than 90 degrees.
2. Choose the **Map | Digitize** command and the pointer becomes a cross hair.
3. Click the left mouse button in the plot window to write the current coordinates to the digitize editor window.
4. If you wish to create multiple polylines, click the digitize editor window and manually insert a blank line between the coordinates that make up each polyline. Each group of contiguous coordinates will form a separate polyline in the blanking file.
5. When you are finished defining the polyline(s) or polygon(s), select the **File | Save As** command in the digitize editor window, and save the file using a [.BLN] extension. Press ESC or choose any button to exit digitize mode.



Two polylines are defined in this window. The first polyline has two vertices and the second has three.

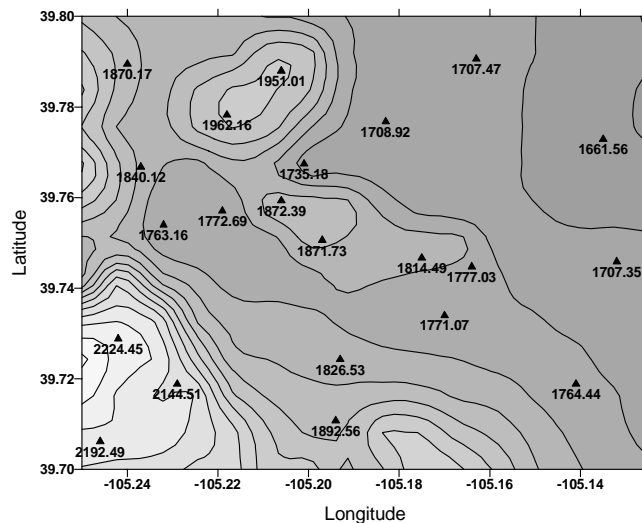
Chapter 8

Post Maps

Introduction to Post Maps

Post maps indicate XY locations with symbols and labels. Posting data points on a map can be useful for determining the distribution and density of your data, as well as placing data or text information at specific locations on the map. You can change the symbol type, size, color, and angle for the data points. An associated data value or text string may be placed next to the posted point. The size, angle, color, and typeface for the label can also be specified.

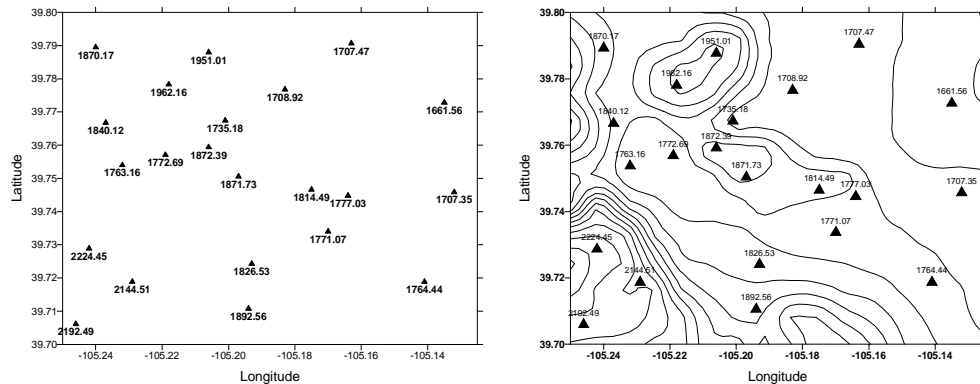
Post maps can be used to show the spatial distribution of the original data when they are overlaid on a grid-based map (such as a contour map). This is often an excellent means of presenting a qualitative measure for the accuracy of the contour lines on the map.



This is an example of a post map overlaid on a filled contour map.

The limits of a post map are based on the limits of the X and Y coordinates contained within the associated data file. These limits can be changed if necessary. See *Chapter 14, Common Map Properties* for more information on limits.

Classed post maps are a type of post map. Post maps display a fixed or proportionally sized symbol at each data location. Classed post maps indicate XY locations with symbols and labels and require Z values in addition to the XY coordinates. Classed post maps group the data into discrete classes (bins). The data points are displayed using the symbol assigned to the class. Classed post maps can also include a legend.



Post maps can stand alone to display the distribution of data points, or can be combined with other maps.

Post Map and Classed Post Map Components

Post maps and classed post maps consist of several different components, each of which can be customized to control the appearance of the map. The components are described briefly here.

- Symbols indicate XY locations.
- Symbol properties include color, size, and angle (post map only). In a post map, gradationally sized symbols indicate data values. In a classed post map, a group of the same type of symbol indicates a range of data values.
- Labels can be added to symbols.
- Map scale controls the size of the map and the number of map units per page unit.
- Map axes display tick marks and axis labels along the borders of the map. Axes show the X and Y scaling in map units. You can include grid lines as extensions of the tick marks over the extent of the map.
- Map orientation is defined as the rotation, tilt, and field of view of the map.
- Map limits are the XY extents of the map.
- A background can be added to a map. The map background consists of a fill underneath the map and a border around the map.

Data Files Used for Posting

Data files used to produce post maps contain the X and Y coordinates used to locate points on the map. They can also provide additional information used to place labels on the map, define the symbols to use, the symbol size, and the angle at which each symbol is drawn. The first row of the data file may optionally contain the title for each column. These titles are used by the dialogs to associate a descriptive name with each column letter. Following the titles are the actual data, organized as one point (record) per row. Columns within the data file contain various properties for the datum point such as its X coordinate, Y coordinate, Z value, etc. Incomplete or blank records are not plotted on the map.

Data files used for post maps and classed post maps can contain slightly different information. Both types of files must contain XY coordinates used to locate the points on the map. In addition, data files for classed post maps must contain a column with data values used to determine the class for each point.

XY Coordinates in the Data File

The X and Y coordinates define the symbol location and the extents of the post map. In the sample file shown below, the X coordinate is in the *Easting* column (Column A), and the Y coordinate is in the *Northing* column (Column B). These coordinates can be in any columns in the data file, but **Surfer** assumes X is in Column A and Y is in Column B by default.

	A	B	C	D	E	F
1	Easting	Northing	Depth	Well Name	Symbol Angle	Symbol Number
2	14326	7753	576	TC-3	30	104
3	14375	7659	489	TC-7	30	104
4	14426	7575	503	WC-5	30	110
5	14537	7283	577	TC-13	45	104
6	13498	7444	589	WC-9	50	110
7	17521	7366	501	TC-8	35	104

A data file used to create a post map can contain several columns of data.

Z Values in the Data File

Z values are the data values, such as elevation or concentration, associated with each XY location. Typically, this is the same column used when creating a grid file. Z values can be used to proportionally scale the posted symbols. You can define the minimum and maximum symbol size, and all points are scaled in proportion to these sizes. See the *Scaled Symbols* section on page 256 for more information.

For classed post maps, the Z value is used to determine which class or bin will contain the data. Classes are based on specified data ranges, and each class is assigned a unique symbol.

A common application for post maps and classed post maps is as overlays on contour maps. The Z values used to scale the posted symbols in a post map, or to define the classes in a class post map, may be the same Z values used to generate the contour map, though this is not required. For example, the posted symbols could be used to indicate rock type, while the contours show ore grade.

Data Labels in the Data File

Data labels are text strings or numbers associated with each point on a post map. Labels may be the original data values for the data points, or may be other identifying text such as well names or sample numbers. Labels can use math text instructions to define custom character formatting. See *Appendix B, Math Text Instructions* for more information.

Symbol Specifications in the Data File

Post map data files can contain a column defining which symbol set and symbol index to use for each posted point. This information can be specified in one of three ways:

<i>SymbolSet:Index</i>	This form allows both the symbol set and the symbol index to be specified. <i>SymbolSet</i> specifies the face name of the desired symbol set. The colon character must appear between the symbol set and the index. If the specified face name is invalid, the default symbol set specified in the post map properties dialog is used instead.
<i>Index</i>	If a single integer is specified, it is interpreted as a symbol index into the current symbol set. The current symbol set is the last specified symbol set or the default symbol set specified in the post map properties dialog.
<i><Empty></i>	If the cell is empty, the last specified symbol set and default symbol index is used.

The symbol index is the symbol or glyph number as it appears in the **Symbol Properties** dialog. This is the 0-based offset of the symbol within the symbol set. However, if the current symbol set is not *Default Symbols*, then 32 must be added to the index value obtained from the **Symbol Properties** dialog. This makes the symbol index the same as its ASCII code. You can use the Window's Character Map utility to determine the ASCII code for font symbols.

	A	B	C	D
1	Easting	Northing	Elevation	Symbol set index
2	4.7	1	66	Ariel:65
3	4.6	1.6	70	Ariel:66
4	4.5	2.5	80	Ariel:67
5	4.6	3.6	95	Default Symbols:4
6	4.5	4.2	80	Default Symbols:8
7	4.3	5.1	70	Default Symbols:14

This is a sample data file containing the symbol set and the index for symbols on a post map.

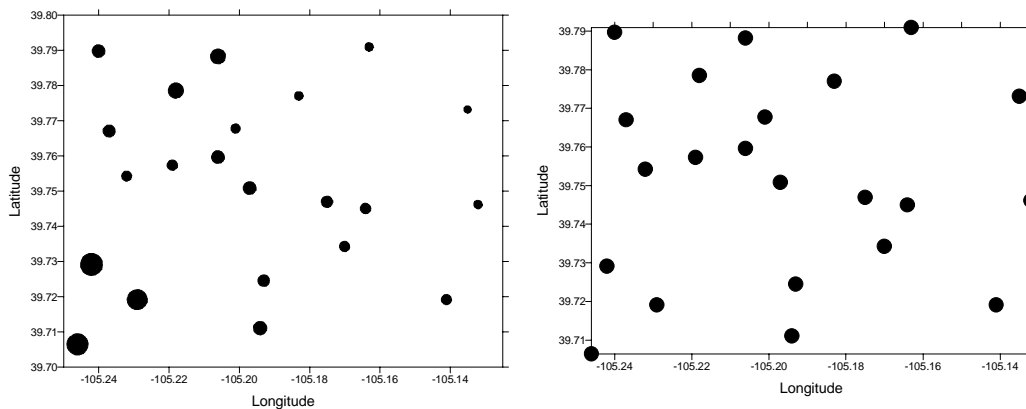
Symbol Angle Values from the Data File

The angle (in degrees) for the posted symbol can be specified in the data file for non-classed post maps. Positive angles rotate the symbols in a counterclockwise direction. On classed post maps, all symbols are drawn at the same orientation, which is specified within the classed post map properties dialog.

Creating a New Post Map or Classed Post Map

To create a post map:


1. Choose the **Map | Post Map | New Post Map** command, or click the  button.
2. Select a data file in the **Open** dialog and then click the *Open* button.
3. The post map is created using the post map dialog default settings.

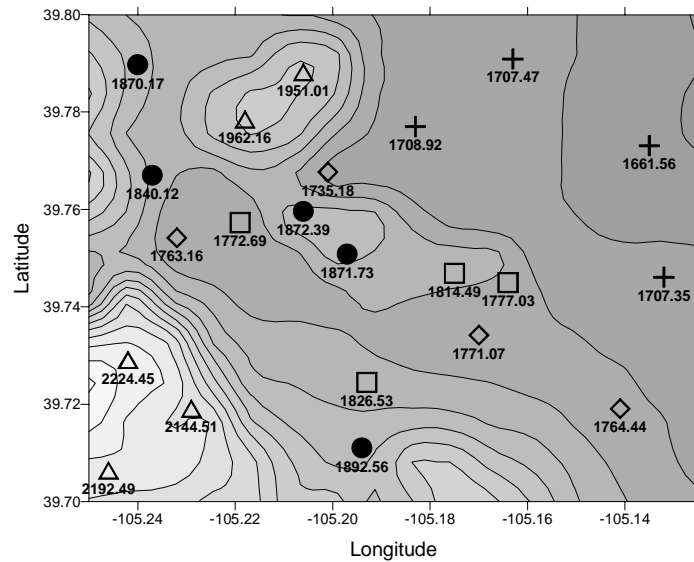


Post maps can contain symbols of all the same size, or scaled proportionally based on data values from the worksheet.

Chapter 8 - Post Maps

To create a classed post map:

1. Choose the **Map | Post Map | New Classed Post Map** command or click the  button.
2. Select a data file in the **Open** dialog and then click the *Open* button.
3. The classed post map is created using the classed post map dialog default settings.



*Classed post maps automatically display
different symbols for different classes of data.*

If the map does not have any symbols in it, then the map was not created with columns containing numerical data. If this is the case, open the map properties and adjust the *X Coord* and *Y Coord* columns.

Editing Post Map or Classed Post Map Properties

To change the features of a post map or classed post map, open the map properties dialog by:

- double-clicking on the map in the plot window,
- double-clicking on the map name in the **Object Manager**,
- right-clicking the map and choosing **Properties** in the context menu,
- or by clicking **Edit | Properties** when the map is selected.


Post Map and Classed Post Map Properties

The post map properties dialog contains six pages: **General**, **Labels**, **View**, **Scale**, **Limits**, and **Background**. The classed post map dialog contains seven pages: **General**, **Labels**, **Classes**, **View**, **Scale**, **Limits**, and **Background**. **View**, **Scale**, **Limits**, and **Background** are discussed in *Chapter 14, Common Map Properties*.

General - Post Map

The post map properties **General** page contains the following options:

Data File

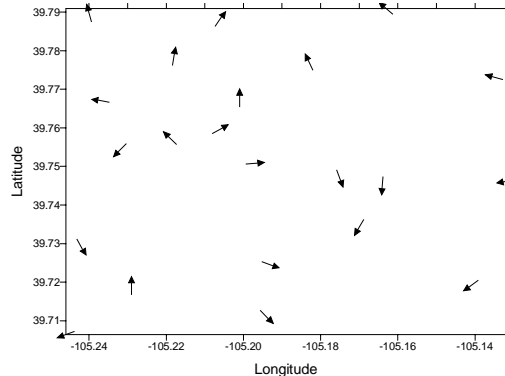
The *Data Filename* group displays the current file used in the post map. Click the  button to display the **Open** dialog. This allows a new or updated data file to be specified for the post map.

Worksheet Columns

The *Worksheet Columns* group specifies the columns to be used from the data file. Set *X Coord* and *Y Coord* to the columns containing the X and Y coordinates respectively. If an empty cell is encountered in either of these columns, the corresponding point is excluded from the map.

Use the *Symbol* list to choose an optional data column containing symbol information.

The *Angle* list allows you to identify an optional column containing the angle, in degrees, by which to rotate each symbol. Positive values rotate the symbol counterclockwise and negative values rotate the symbol clockwise. If an empty cell is encountered, the *Default Angle* is used for that point.



A post map using angle values read from the data file. This allows you to post symbols at any angle.

Chapter 8 - Post Maps

Default Symbol

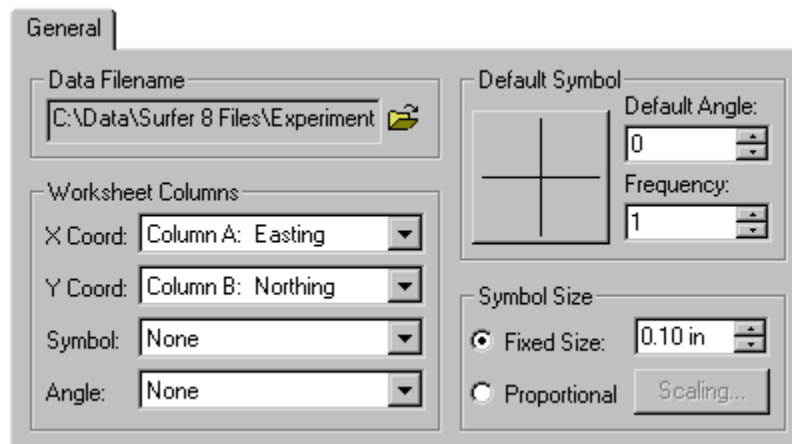
The *Default Symbol* button is used to specify the symbol set and symbol (unless overridden by values in the *Symbol* column). Click the button displaying the current symbol to open the **Symbol Properties** dialog. From here, you can change the symbol set, symbol index, and symbol color. To eliminate the display of symbols, select one of the blank symbol positions in the symbol palette.

The *Default Angle* specifies the angle to apply to all the symbols on the map (unless overridden by values in the *Angle* column). Positive angles, in degrees, rotate the symbol in a counterclockwise fashion. Negative angle values, in degrees, rotate the symbol clockwise.

Use the *Frequency* setting to control how often data points are posted. A frequency of 1 posts every point. A frequency of 2 posts every other point, etc. This is often used to thin the number of displayed data points to avoid overwriting each other.

Symbol Size

The *Symbol Size* group allows you to specify the symbol size in two ways: fixed or proportional. *Fixed Size* uses the same size symbols throughout the map. The *Proportional* option sizes all symbols proportionally, based on scaling information in the **Proportional Scaling** dialog. See the *Scaled Symbol* section on page 256 for more information on the **Proportional Scaling** dialog.




*You can use the **General** page to change the data file, set the worksheet columns used in the plot, and set symbol properties.*

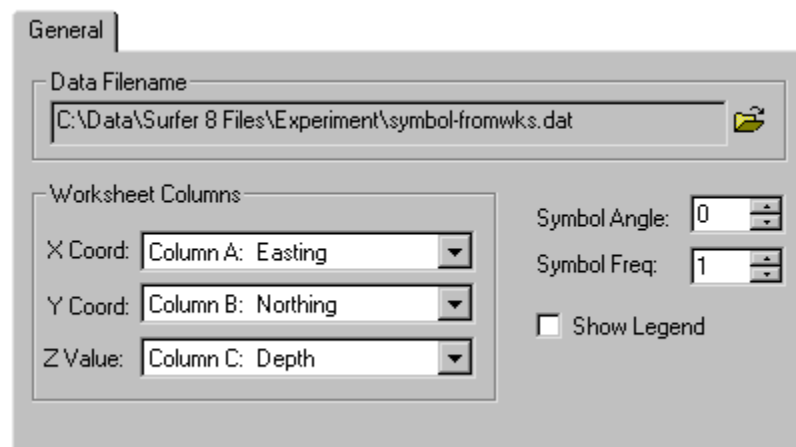
General - Classed Post Map

The classed post map properties **General** page contains the following options:

Data File

The *Data Filename* group displays the current file used in the classed post map.

Click the  button to display the **Open** dialog. This allows a new or updated data file to be specified for the classed post map.



Worksheet Columns

*You can use the **General** page to change the data file, set the worksheet columns used in the plot, and to create a legend.*

The Worksheet

Columns group specifies the columns to be used from the data file. Set *X Coord* and *Y Coord* to the columns containing the X and Y coordinates respectively. Set the *Z Value* to the column containing the values that will be used to determine the classes. If an empty cell is encountered in any of these columns, the corresponding point is excluded from the map.

Symbol Angle

The *Symbol Angle* specifies the angle, in degrees, to apply to all the symbols on the map. Positive angles rotate the symbol in a counterclockwise fashion and negative angle values rotate the symbol clockwise.

Symbol Frequency

Use the *Symbol Freq* setting to control how often data points are posted. A frequency of 1 posts all data. A frequency of 2 posts every other point, etc. This is often used to thin the number of displayed data points to avoid overwriting each other.

Legend

Check the *Show Legend* option to display a legend that relates the symbols in the map to each class.

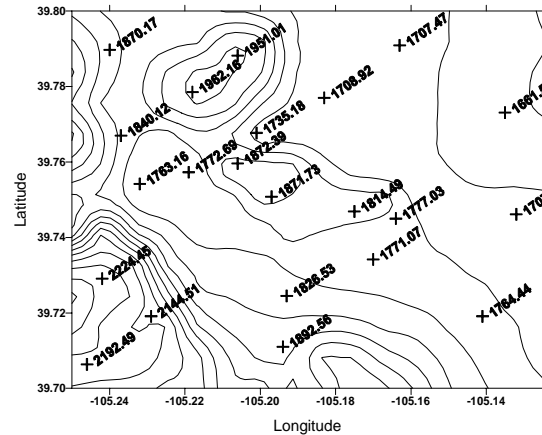
Chapter 8 - Post Maps

Labels

When posting data points on a map, you can associate text from the worksheet with each posted symbol. The position, font, and numeric format can also be specified. Both post maps and classed post maps use the same **Labels** page.

Worksheet Label Column

Select the worksheet column containing the values or text you wish to label the posted points with in the *Worksheet Column for Labels* list. Labels may be the original data values for the data points, or may be other identifying text such as well names or sample numbers. Labels can use math text instructions to define custom character formatting.



The label position and angle can be changed on the map. In this example, the labels are rotated and positioned to the right of the data points.

Position Relative to Symbol

The *Position Relative to Symbol* group controls the offset of the label from the symbol center. You can select *Center*, *Left*, *Right*, *Above*, *Below*, or *User Defined* positions. *User Defined* allows you to specify the exact offset (in page units) in the *X Offset* and *Y Offset* boxes. The posted labels are all placed in the same position relative to the associated symbol. Positive values shift the label position to the right and upward.

Label Angle

The *Angle* box specifies the angle (in degrees) at which to draw the labels. Positive angles rotate the symbols counterclockwise.

Label Plane

The *Plane* list specifies the coordinate plane that contains the label. If *X-Y Plane* is selected, labels are oriented parallel to the XY plane of the map. As the map is tilted in 3D, the labels tilt as well. This can make it hard to read the labels at shallow tilt angles since they are viewed on edge. In this case, it may be better to orient the labels in the plane of the *Screen*. Screen-oriented labels are always displayed perpendicular to the viewer's line of sight, no matter how the map is tilted or rotated.

Font

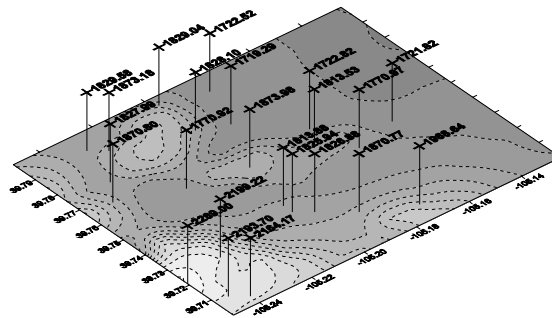
You can specify text properties for the label by clicking on the *Font* button.

Format

Click the *Format* button to display the **Label Format** dialog. The numeric format is applied to all numbers read from the specified label column.

3D Label Lines

3D Label Lines are used when the post map or classed post map is tilted to any angle other than 90 degrees. In this case, the 3D label lines define the distance above the map that the points are posted. This option has no effect when the post map is tilted at 90 degrees. The *Length* box specifies how long the label lines are and how far above the surface the labels are drawn.



3D Label Lines can be used when you tilt a map to any angle other than 90 degrees. On this map, 3D Label Lines are used to post points on a tilted filled contour map.

3D label lines are not drawn on post maps overlaid on surfaces.

You can click the *Properties* button to display the **Line Properties** dialog. This dialog is used to specify the properties for the *3D Label Lines*.

*The **Labels** page is used to select the worksheet column containing labels and to set the label properties.*

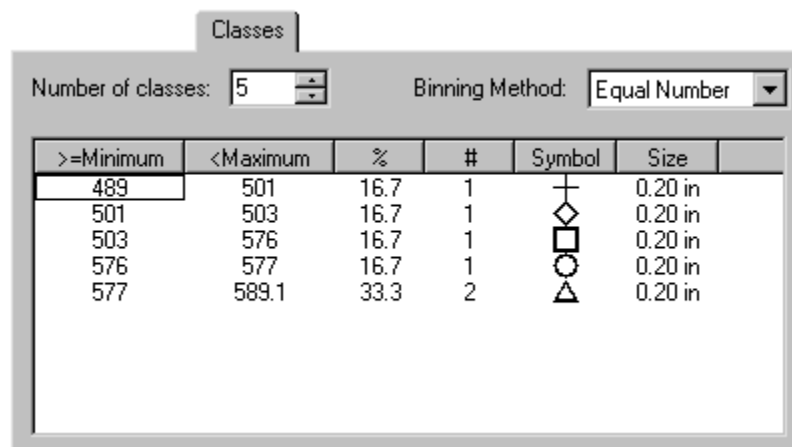
Classes

The classed post map **Classes** page allows you to define the classes used to group the data values. Each class is represented by a unique symbol in the classed post map.

The *Number of classes* value is used to specify how many classes or groupings there will be on the map. When you change the *Number of classes* value, the class list box is automatically updated to reflect the change.

The *Binning Method* specifies the method used to calculate the limits of the classes:

- *Equal Number* assigns the class ranges such that approximately equal numbers of points are included in each class. In this case, the interval of each class is usually different.
- *Equal Intervals* assigns the class ranges so the interval between the \geq *Minimum* value and the $<$ *Maximum* value is equal for each class. In this case, different numbers of points might be assigned to each class.
- *User Defined* allows you to set the \geq *Minimum* value and the $<$ *Maximum* value for each class. This allows you to specify your own ranges for the classes. Ranges defined in this way do not have to be contiguous. To change the \geq *Minimum* or $<$ *Maximum* value, you can double-click the values in the class list.

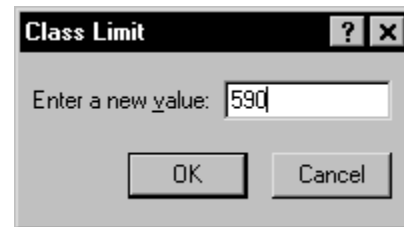


\geq Minimum	$<$ Maximum	%	#	Symbol	Size
489	501	16.7	1	+	0.20 in
501	503	16.7	1	◇	0.20 in
503	576	16.7	1	□	0.20 in
576	577	16.7	1	○	0.20 in
577	589.1	33.3	2	△	0.20 in

*You can set the binning method and the bin ranges on the classed post map **Classes** page.*

The class list box displays summary statistics and allows you to specify the properties for each class.

- The \geq *Minimum* list specifies the lower limit for each class of data. You can double-click the number for any of the classes and change the value in the **Class Limit** dialog.
- The $<$ *Maximum* list specifies the upper limit for each class of data. You can double-click the number for any of the classes and change the value in the **Class Limit** dialog.
- The % column indicates the percentage of data points in the particular class. This value cannot be edited and is for informational purposes only.
- The # column indicates the number of points included in each class. This value cannot be edited and is for informational purposes only.
- The *Symbol* column displays the symbol used for each class. To change a symbol or symbol property used for a particular class, double-click the symbol, and then make changes in the **Symbol Properties** dialog.
- The *Size* column specifies the size of the symbol. To change the size of a symbol for a particular class, double-click the *Size* value in the list, and change the value in the **Symbol Properties** dialog.




*To change a bin limit, double-click on the limit value to open the **Class Limit** dialog. Enter the new value into the **Class Limit** dialog.*

Updating the Data File

A copy of the data file is created and embedded within the post map at the time the post map is created. This means any subsequent changes made to the data file are not reflected in the post map. To incorporate changes made to a data file into its associated post map, the data file must be reassigned to the map.

To update a post or classed post map:

1. Select the map to be updated.
2. Open the map properties using one of the methods listed in the *Editing Post Map or Classed Post Map Properties* section.
3. On the **General** page, click on the  button in the *Data Filename* group.
4. Select the data file with the incorporated changes and click the *Open* button.
5. Click OK or *Apply* to redraw the map with the updated data.

Classed Post Map Legends

Legends can be displayed for classed post maps.

Display Legend

Check the *Show Legend* option in the **General** page of the classed post map properties dialog to display a legend that relates the symbols in the map to each class.

Edit Legend Properties

To change the legend properties, double-click the legend. The **Legend Properties** dialog is displayed.

Legend Properties Dialog

Double-click on the legend to open the **Legend Properties** dialog.

Frame

You can use the *Frame* group to set the properties to use for the border around the legend. The *Type* list specifies the type of border to use for the legend. You can select *None*, *Square*, or *Rounded* for the frame type. The *Margin* specifies the distance between the frame and the text or symbols within the legend. You can type a margin value or click the up and down arrows to specify the *Margin* distance. The *Line* button displays the **Line Properties** dialog. From there, you can specify the line properties to use for the legend frame. The *Fill* button displays the **Fill Properties** dialog. The fill properties to use for the background of the legend frame may be specified here.

Title

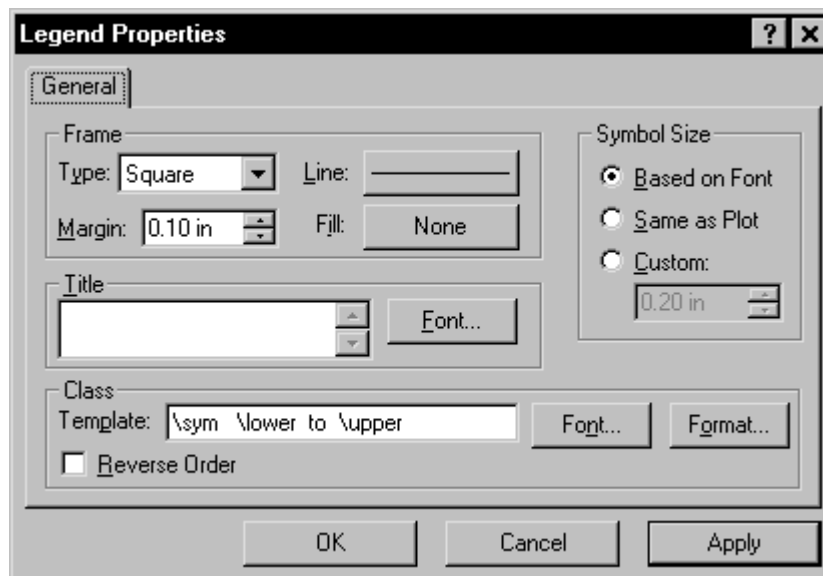
The *Title* group allows you to create a title to use on the legend. You can type a title into the box. The title may include math text instructions. To add new lines in the title, use CTRL+ENTER. Click the *Font* button to set the title font properties.

Class

The *Class* group is used to specify how the different classes are represented in the legend. This controls the type of information placed in the legend, and how the information is formatted.

Classed Post Map Legend	
+	347 to 461
◇	461 to 490
□	490 to 511
●	511 to 533
△	533 to 616.1

A classed post map legend shows the symbols and their associated data ranges on a classed post map.



You can open the **Legend Properties** dialog by double clicking on the legend.

Template

The *Template* box allows you to customize the format of the legend items. The template is applied to each class. Normal text and math text instructions can be included within the template in addition to the following special template directives:

<code>\sym</code>	places the class symbol in the legend
<code>\lower</code>	places the lower class limit in the legend
<code>\upper</code>	places the upper class limit in the legend
<code>\tab</code>	places a tab in the class text, tab stops are defined at every 0.25 inches starting at the frame edge

For example, the template text:

`\sym >= \lower \tab < \upper \tab`

\triangle	≥ 0.54	< 74.66
\diamond	≥ 74.66	< 90.78

yields the legend text shown in the example at the right. Any number of directives can be included in any order.

Chapter 8 - Post Maps

Click the *Font* button in the *Class* group to display the **Text Properties** dialog. From here, you can specify the text properties to use for the class text. Use the *Format* button to display the **Label Format** dialog. This dialog allows you to set the numeric format used by numbers in the in the \lower and \upper directives.

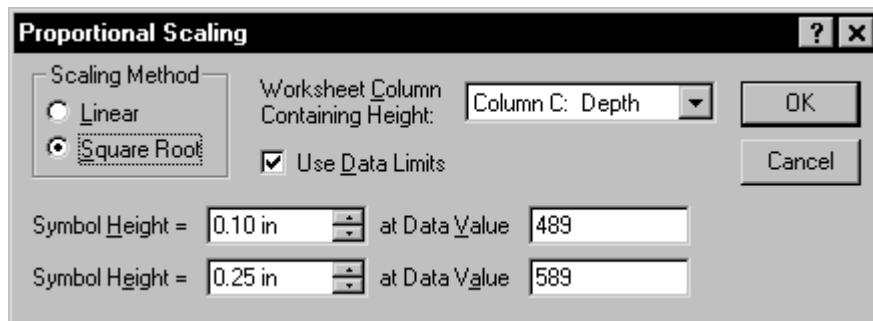
The *Reverse Order* check box reverses the order of the class list display. When the *Reverse Order* check box is activated, the class with the largest values is at the top of the list. When not activated, the class with the smallest values is at the top of the list.

Symbol Size

The *Symbol Size* group allows you to specify the size of the legend symbols. Selecting *Based on Font* causes the symbols to be drawn using the size specified in the *Class* group. Selecting *Same as Plot* draws the symbols the same size as they appear on the map. Selecting *Custom* allows a size to be entered for the legend symbols in the box below the *Custom* option.

Scaled Symbols

The *Symbol Size* group in the **General** page of the post map properties dialog allows you to specify the symbol size in two ways: fixed or proportional. *Fixed Size* uses the same size symbols throughout the map. Click the *Fixed Size* option button and set the size in page units.



The *Proportional Scaling* dialog is opened by clicking on the *Scaling* button on the **General** page in the post map properties dialog.

The *Proportional* option sizes all symbols proportionally, based on scaling information in the **Proportional Scaling** dialog. By default, the scaling is based on the data in the third-used column. This column usually contains the Z values. If you want to scale your symbols based on data in a different column, click the *Scaling* button to display the **Proportional Scaling** dialog. You can select the symbol scaling column in the *Worksheet Column Containing Height* list.

For proportional scaling, **Surfer** uses two *Symbol Height / Data Value* pairs to define the scaling equations. These pairs define the symbol size at two different data values, usually the minimum and maximum values in the height column. All values in between are scaled proportionally. Assume the first symbol height is defined as H_1 and is associated with the minimum Z value in the data file (Z_1). Similarly, the maximum symbol height is defined as H_2 and is associated with the maximum Z value in the data file (Z_2). Then H_n is the symbol height for a symbol with the proportional value Z_n .

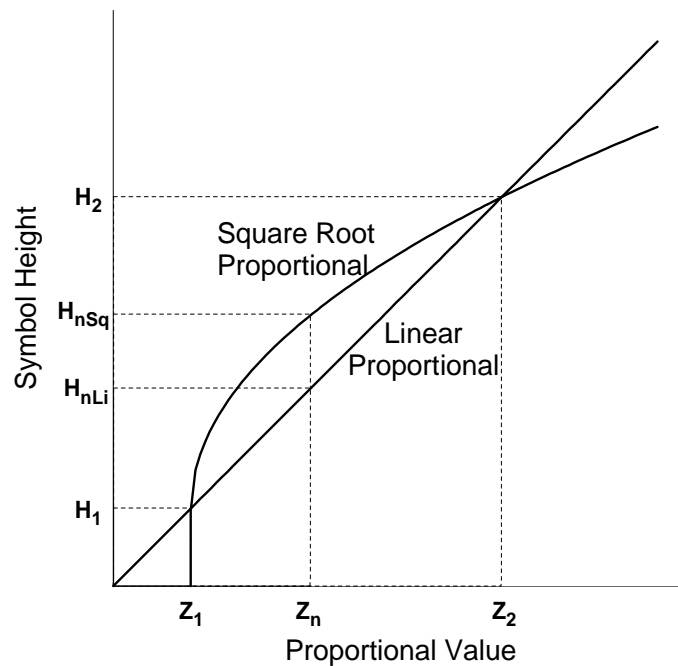
If *Scaling Method* is set to *Linear*, a standard linear interpolation is used:

$$H_n = [((Z_n - Z_1)/(Z_2 - Z_1)) * (H_2 - H_1)] + H_1$$

If *Scaling Method* is set to *Square Root*, the equation used is:

$$H_n = \left[\sqrt{(Z_n - Z_1)/(Z_2 - Z_1)} \times (H_2 - H_1) \right] + H_1$$

Square root scaling is commonly used with solid symbols to offset the fact that the area increases as a function of the symbol height squared. Square root scaling is essentially making the area of the symbol proportional to the Z value, rather than making the size of the symbol proportional to the Z value.



This graph shows the height versus Z value relationship for linear proportional symbols and square root proportional symbols.

Enter the values you wish to use for H_1 , H_2 , Z_1 , and Z_2 into the *Symbol Height* and *Data Value* fields respectively. Check the *Use Data Limits* option to use the minimum and maximum data values in the specified height column for Z_1 , and Z_2 .

Chapter 9

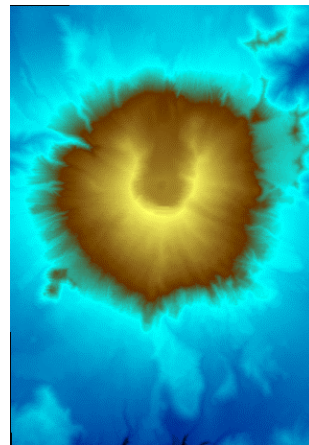
Image Maps

Introduction to Image Maps

Image maps represent grid files as raster (bitmap) images. In a raster image, each pixel over the mapped area is independently assigned a color. Image maps represent different Z-values with color.

The color schemes for image maps can be saved in color spectrum files [.CLR]. With such a file, the colors defined for one image map can be used with any other image map. Since the color anchor points are stored as a percentage of the grid data range, a single color spectrum file can be used for multiple maps, even if the associated grid files cover significantly different data ranges.

Surfer ships with a wide variety of predefined [.CLR] files in the SAMPLES directory.



This is an image map of Mt. St. Helens after the 1980 eruption. The image map represents different elevations with different colors.

USGS Elevation Files

The elevation data contained in USGS digital elevation model files are regularly spaced, but the direction of the grid lines may not exactly coincide with the grid edges. This is due to the variable angle between the UTM coordinate system (used in some DEM files) and true north. This results in the potential for a regular, stair-stepped arrangement of the grid nodes along the edges of the map that can translate to blanked grid nodes along the edges of maps produced in **Surfer**. If the DEM file contains three-second arc coordinates, the coordinates are automatically converted to latitude/longitude coordinates in **Surfer**.


Image Map Components

Image maps consist of several different components, each of which can be customized to control the appearance of the map. The components are described briefly here.

- Color indicates Z information and is controlled through color spectrum files [.CLR].
- Color scale bars can be added to show the relation between color and Z values.
- Map scale controls the size of the map and the number of map units per page unit.
- Map axes display tick marks and axis labels along the borders of the map. Axes show the X and Y scaling in map units. You can include grid lines as extensions of the tick marks over the extent of the map.
- Map orientation is defined as the rotation, tilt, and field of view of the map.
- Map limits are the XY extents of the map.
- A background can be added to a map. The map background consists of a fill underneath the map and a border around the map. Typically, only the background line color is visible on image maps.

Creating a New Image Map

To create an image map:

1. Click **Map | Image Map** or click the  button.
2. Select a grid file in the **Open Grid** dialog and click the *Open* button.
3. The map is created.

Editing Image Map Properties

To change the features of the image map, open the image map properties dialog by:

- double-clicking on the map in the plot window,
- double-clicking on the map name in the **Object Manager**,
- right-clicking the map and choosing **Properties** in the context menu,
- or by clicking **Edit | Properties** when the image map is selected.

Image Map Properties


The image map properties dialog contains five pages: **General**, **View**, **Scale**, **Limits**, and **Background**. **View**, **Scale**, **Limits**, and **Background** are discussed in *Chapter 14, Common Map Properties*.


General

The image map properties **General** page contains the following options:

Input Grid

The *Input Grid* group indicates the path and file name for the grid file used for the map.

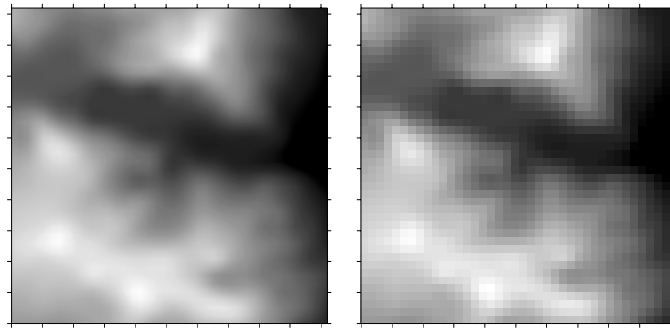
Click the  button to display information about the grid file used to produce the image map. This shows the number of rows and columns in the grid, the minimum and maximum X, Y, and Z values in the grid file, and statistics. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.

Click the  button to change the grid file. The **Open Grid** dialog is displayed, where a different grid file can be selected.

Interpolate Pixels

The *Interpolate Pixels* check box activates color smoothing on the map. When activated, *Interpolate Pixels* uses bilinear interpolation to calculate colors on the map. Bilinear interpolation makes the color gradations smoother, but it can slightly slow the on-screen drawing of the image map.

When a dense grid (a grid with relatively large numbers of rows and columns) is used, little difference is seen between the final image maps whether the *Interpolate Pixels* option is selected or not. For dense grids, therefore, on-screen drawing time can be reduced when the *Interpolate Pixels* check box is off.

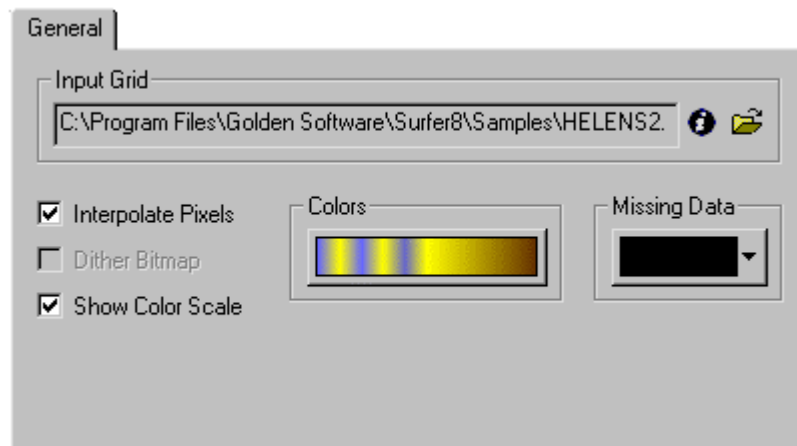


*These are two image maps of the same grid file. The *Interpolate Pixels* option has been selected for the map on the left. This option has not been selected for the map on the right, and the grid cells are quite evident.*

When a coarse grid (a grid with relatively few rows and columns) is used and the *Interpolate Pixels* option is not selected, all pixels within a single grid square are assigned the same color (nearest neighbor interpolation). This creates a grid-square map, and can result in an image map with a very blocky appearance. For coarse grids, therefore, a smoother appearance results when the *Interpolate Pixels* check box is activated.

Dither Bitmap

The *Dither Bitmap* check box is used to enable dithering on displays limited to 256 colors. Dithering provides a smoother image when there are a limited number of colors available. Dithering can lengthen redraw time. This option is grayed if your monitor is capable of displaying more than 256 colors.



*The image map properties **General** page controls the image map color and color scale bar.*

Color Scale Bar

The *Show Color Scale* option permits the display of a color scale bar adjacent to the map. The color scale bar indicates the colors assigned to the Z levels on the map, and the associated Z level values are displayed as labels on the color scale bar. When the *Show Color Scale* check box is selected, the color scale bar is displayed.

Colors

The *Colors* group shows the spectrum of colors used to draw the map. To change the colors, click on the *Colors* button to open the **Color Spectrum** dialog. See *Chapter 16, Adding Color to Maps* for more information on the **Color Spectrum** dialog.

Missing Data

The *Missing Data* group shows the color drawn for blank regions of the grid. To select a different color, click on the arrow button to the right of the sample and choose a color from the palette. Click the *Custom* button at the bottom of the color palette to define new colors.

Image Map Overlays

Image maps can be created independently of other maps, or they can be combined with other maps. Image maps cannot be overlaid on wireframes, but image maps can be overlaid on surfaces and other map types.

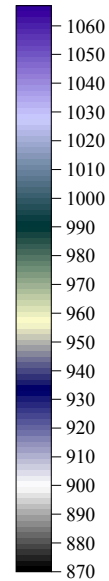
Color Scale Bars

A color scale bar is a legend that shows the colors used in an image map.

Adding a Color Scale Bar

To add a color scale bar to an image map:

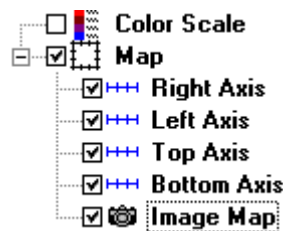
1. Open the map properties dialog using one of the methods listed in the *Editing Image Map Properties* section of this chapter.
2. Specify the colors to use in the map on the **General** page.
3. Check the *Show Color Scale* box to activate the color scale bar.
4. Click OK or *Apply* to draw the map with a color scale bar.



Color scale bars are added through the image map properties dialog.

Color Scale Bar Visibility

The color scale bar can be toggled on and off through the **Object Manager**.



*Remove the check mark from the Color Scale box in the **Object Manager** to make the color scale bar invisible.*

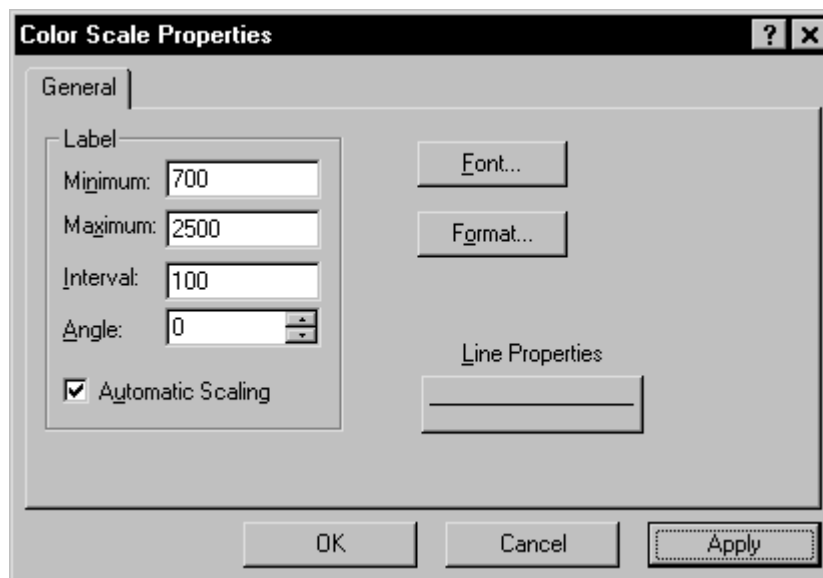
Removing a Color Scale Bar

A color scale bar can be removed by clicking on the scale bar and pressing the DELETE key on your keyboard. Alternatively, you can uncheck the *Show Color Scale* check box in the map properties dialog.

Editing a Color Scale Bar

To modify the color scale bar:

1. Double-click the color scale bar to display the **Color Scale Properties** dialog.
 - Set the first label value in the *Minimum* box.
 - Set the last label value in the *Maximum* box.
 - Define the spacing between labels in the *Interval* box.
 - Choose the angle at which the labels are drawn in the *Angle* box. Positive angles rotate the labels in a counterclockwise direction.
 - The *Automatic Scaling* check box sets the label frequency so labels do not overlap. If there are too many labels to fit into the space next to the color scale bar, some labels are dropped to make the labels legible.
 - Click the *Font* button to set the label text properties.
 - Click the *Format* button to choose the label numeric format (i.e. number of decimal places).
 - The *Line Properties* button allows you to specify the line properties for the color scale bar border.
2. Once the changes are made, click OK or *Apply* to redraw the color scale bar.



The *Color Scale Properties* dialog controls the display of the color scale bar.

Chapter 10

Shaded Relief Maps

Introduction to Shaded Relief Maps

Shaded relief maps are raster (bitmap) maps based on grid files. These maps use colors to indicate the local orientation of the surface relative to a user-defined light source direction. **Surfer** determines the orientation of each grid cell and calculates reflectance of a point light source on the grid surface. The light source can be thought of as the sun shining on a topographic surface. Portions of the surface that face away from the light source reflect less light toward the viewer, and thus appear darker.

Reflectance

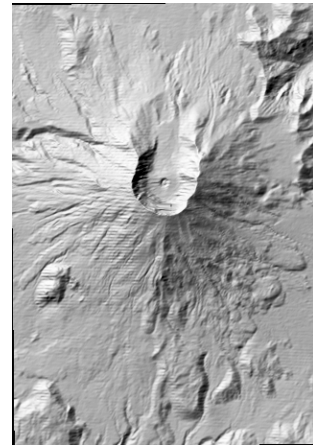
The colors on a shaded relief map are based on the reflectance from the grid surface. Reflectance values range from zero to one. A reflectance value of zero means that no light is reflected toward the viewer. A reflectance value of one means that all incident light is reflected toward the viewer.

Colors

Because colors are assigned to entire grid cells, sparse grids (grids with relatively few rows and columns) are poor candidates for shaded relief maps. Shaded relief maps based on grids with too few cells look blocky or fuzzy.

You can assign colors to represent the various reflectance values. Colors between the assigned values are automatically blended to make a smooth gradation. Blanked regions on shaded relief maps are shown as a separate color.

Any color scheme chosen for a shaded relief map may be used with any other shaded relief map, even if the associated grid files result in significantly different orientations of the surface relative to the light source. Color schemes can be saved in color spectrum files [.CLR]. **Surfer** ships with a wide variety of [.CLR] files in the SAMPLES directory.



This is a shaded relief map of Mt. St. Helens after the 1980 eruption. The shaded relief map represents different intensities of reflected light with different shades of gray; the light source is located to the northwest.

USGS Elevation Files

The elevation data contained in USGS digital elevation model files are regularly spaced, but the direction of the grid lines may not exactly coincide with the grid edges. This is due to the variable angle between the UTM coordinate system (used in some DEM files) and true north. This results in the potential for a regular, stair-stepped arrangement of the grid nodes along the edges of the map that can translate to blanked grid nodes along the edges of maps produced in **Surfer**. If the DEM file contains three-second arc coordinates, the coordinates are automatically converted to latitude/longitude coordinates in **Surfer**.


Shaded Relief Map Components

Shaded relief maps consist of several different components, each of which can be customized to control the appearance of the map. The components are described briefly here.

- Color is controlled through color spectrum files [.CLR].
- Light source settings can be changed to any direction on the map.
- Map scale controls the size of the map and the number of map units per page unit.
- Map axes display tick marks and axis labels along the borders of the map. Axes show the X and Y scaling in map units. You can include grid lines as extensions of the tick marks over the extent of the map.
- Map orientation is defined as the rotation, tilt, and field of view of the map.
- Map limits are the XY extents of the map.
- A background can be added to a map. The map background consists of a fill underneath the map and a border around the map. Typically, only the background line color is visible on shaded relief maps.

Creating a New Shaded Relief Map

To create a new shaded relief map:

1. Click **Map | Shaded Relief Map** or click the  button.
2. Select a grid file in the **Open Grid** dialog and click the *Open* button.
3. The map is created.

Editing Shaded Relief Map Properties

To change the features of the shaded relief map, open the shaded relief map properties dialog by:

- double-clicking on the map in the plot window,
- double-clicking on the map name in the **Object Manager**,
- right-clicking the map and choosing **Properties** in the context menu,
- or by clicking **Edit | Properties** when the shaded relief map is selected.

Shaded Relief Map Properties


The shaded relief map properties dialog contains five pages: **General**, **View**, **Scale**, **Limits**, and **Background**. **View**, **Scale**, **Limits**, and **Background** are discussed in *Chapter 14, Common Map Properties*.


General

The shaded relief map **General** page contains the following options:

Input Grid File

The *Input Grid File* group indicates the path and file name for the grid file used for the map.

You can click the  button to display information about the grid file used to produce the shaded relief map. This shows the number of rows and columns in the grid, the minimum and maximum X, Y, and Z values in the grid file, and statistics. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.

Click the  button to change the grid file. The **Open Grid** dialog is displayed, where a different grid file can be selected.

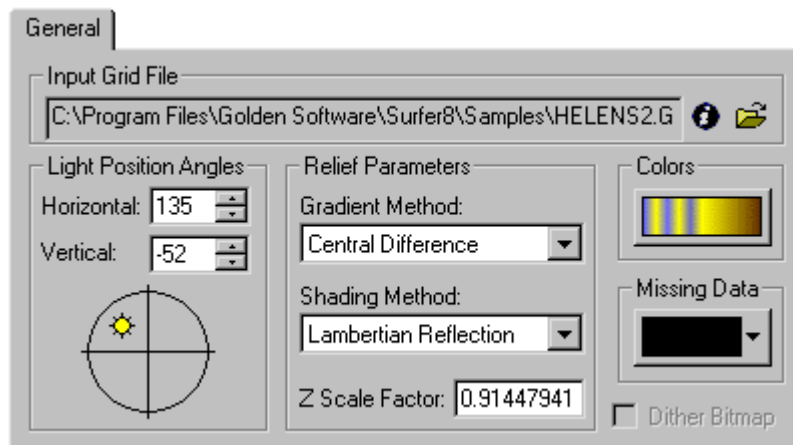
Chapter 10 - Shaded Relief Maps

Light Position Angles

The *Light Position Angles* group permits you to specify the orientation of the light source. The light source can be thought of as the sun shining on a topographic surface. The graphic image shows the relative position of the light source.

The *Horizontal* box defines the direction for the light source in the horizontal plane. Zero degrees corresponds to the light source shining from due east. Positive angles rotate the light source counterclockwise. For example, a specified horizontal angle of 90 degrees places the light source due north of the unrotated surface. The default horizontal angle is set at 135 degrees, or NW.

The *Vertical* box rotates the light source in the vertical plane. A vertical angle of zero degrees places the light source at the horizon. An angle of 90 degrees places the light source directly overhead. You can also specify negative angles, where the light source originates from under the surface. The default vertical angle is 45 degrees. As the vertical angle approaches zero, shadows lengthen and the overall display shifts to the colors at the left end of the color spectrum.



*The shaded relief map **General** page controls the map color and lighting.*

Relief Parameters

The *Gradient Method* controls the smoothness of the shaded relief map.

- The *Central Difference* method averages the slope and orientation of the surface across three adjacent grid nodes. This results in a smoother shaded relief surface. Because a node on either side is required, edge cells are blanked. Blanked cells are assigned the *Missing Data* color.
- The *Midpoint Difference* method computes the gradient at the center of each grid cell. This method provides less smoothing, but does not blank the grid cells at the edge of the map.

The *Shading Method* specifies the algorithm used to compute the reflected light at each grid cell. The *Shading Method* determines how the colors are distributed in relation to the slopes and slope directions (aspect) over the extent of the map. There are four shading methods available for shaded relief maps.

- *Simple* is the fastest of the shading methods, but produces a rather crude image. With *Simple*, the *Horizontal* (azimuth) and *Vertical* (zenith) values are fixed at 135 and 45 degrees, respectively, and cannot be changed.
- *Peucker's Approximation* uses a piecewise linear approximation. This method gives somewhat better results than the *Simple* method, but redrawing the map takes slightly longer. With *Peucker's Approximation*, the *Horizontal* (azimuth) and *Vertical* (zenith) values are fixed at 135 and 45 degrees, respectively, and cannot be changed.
- *Lambertian Reflection* assumes an ideal surface that reflects all the light that strikes it and the surface appears equally bright from all viewing directions. The Light Position Angles *Horizontal* and *Vertical* box values can be specified with this option.
- The *Lommel-Seeliger Law* method is based on an analysis of light scattering from a surface. This method is a compromise between an ideal diffuser and a real surface. With some surfaces, this may actually give better results than the *Lambertian Reflection* method. The *Light Position Angles Horizontal* and *Vertical* box values can be specified with this option.

The *Z Scale Factor* box allows you to specify the Z scale to use for the surface. As you increase the *Z Scale Factor*, the surface becomes more exaggerated vertically. Increasing the factor enhances the shading effect, and can be useful for bringing out more detail, especially on relatively flat surfaces. The default *Z Scale Factor* scales the Z coordinates to 1/10th the XY diagonal of the input grid.

Colors

The *Colors* group shows the spectrum of colors used to draw the map. To change the colors, click on the *Colors* button to open the **Color Spectrum** dialog. Refer to *Chapter 16, Adding Color to Maps* for more information on the **Color Spectrum** dialog.

Chapter 10 - Shaded Relief Maps

Missing Data

The *Missing Data* group button displays the color drawn for blank regions of the grid. To select a different color, click the button and choose a color from the palette. You may click the *Custom* button at the bottom of the color palette to define new colors not available on the palette.

Dither Bitmap

The *Dither Bitmap* check box is used to enable dithering on displays limited to 256 colors. Dithering provides a smoother image when there are a limited number of colors available. Dithering can lengthen redraw time. This option is grayed if your monitor is capable of displaying more than 256 colors.

Shaded Relief Map Overlays

Shaded relief maps can be created independently of other maps, or they can be combined with other maps. Shaded relief maps cannot be overlaid on wireframes, but they can be overlaid on surfaces and other map types.

Reference

Many of the parameters for the shaded relief map are technical in origin, and a detailed development is beyond the scope of this documentation. For a complete presentation of the underlying theory and mathematics, see Horn (1982).

Horn, B.K. (1982), *Hill Shading and the Reflectance Map*, Geo-Processing, v. 2, p. 65-146.

Chapter 11

Vector Maps

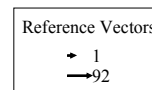
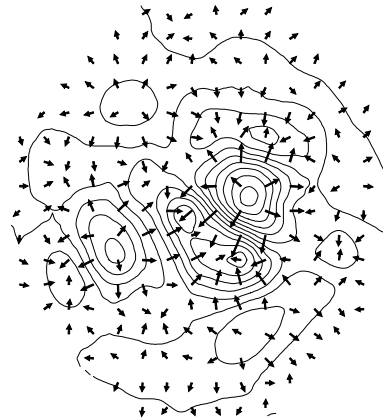
Introduction to Vector Maps

A vector map shows direction and magnitude of data at points on a map. You can create vector maps from information in one grid in **Surfer**, or you can use two grids to generate a vector map.

1-Grid Vector Maps

Vector map information, direction and magnitude, can be derived from one grid. The arrow symbol points in the "downhill" direction and the length of the arrow depends on the magnitude, or steepness, of the slope. A vector is drawn at each grid node unless some nodes are skipped by changing the *Frequency* setting on the **Symbol** page in the vector map properties dialog.

For example, consider a grid containing elevation information. If water were poured over the surface, the direction arrows would point in the direction water flows - from high elevation to low elevation. Magnitude is indicated by arrow length. In the water flow example, the steeper slopes would have longer arrows.



This is a one-grid vector map overlaid on a contour map. The vectors point downhill and the longest vectors indicate the steepest slopes.

2-Grid Vector Maps

Two-grid vector maps use two separate grid files to determine the vector direction and magnitude. The grids can contain Cartesian or polar data. With Cartesian data, one grid consists of X component data and the other grid consists of Y component data. With polar data, one grid consists of angle information and the other grid contains length information.

Vector Map Components

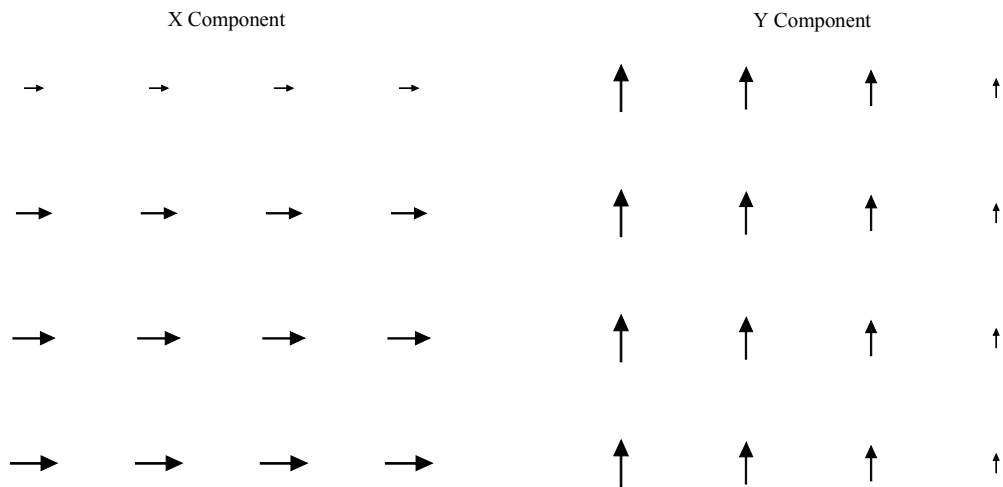
- An arrow represents a vector's direction and magnitude at a grid node.
- The arrowhead points in the downhill direction.
- The length of the arrow represents the steepness or magnitude of the slope.
- You can use color to represent magnitude.
- A color scale bar can show a variable from another grid.
- Color scale bars and vector legends show length and magnitude information for the map.
- Map scale controls the size of the map and the number of map units per page unit.
- Map axes display tick marks and axis labels along the borders of the map. Axes show the X and Y scaling in map units. You can include grid lines as extensions of the tick marks over the extent of the map.
- Map orientation is defined as the rotation, tilt, and field of view of the map.
- Map limits are the XY extents of the map.
- A background can be added to a map. The map background consists of a fill underneath the map and a border around the map.

Two-Grid Vector Map Data

The two components of a vector map, magnitude and direction, can also be created from two grids. You can use either Cartesian or polar data with two-grid vector maps. Both grids must have the same grid line geometry. See *Chapter 4, Creating Grid Files* for more information on grid line geometry.

Cartesian Data

Vector maps created with Cartesian data requires that one grid contain X component data and the other grid contain Y component data. The data from the two grids are combined to produce vector direction and magnitude.

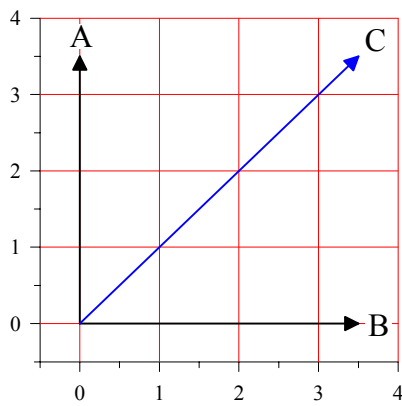


The maps above show two grid files. The left map displays an X component grid and the right map shows a Y component grid. These maps both show X and Y components in positive space, though the components can be in either negative or positive space. The direction and magnitude of the resulting vectors are determined from the two grids as shown in the following example.

Chapter 11 - Vector Maps

Example

The lower right corner's X component is 3.5 and the Y component is 3.5. To simplify the math, the origin of both is zero. The direction of the resulting vector is found by adding the two vectors together:



$$A = (a_1, a_2) = (0, 3.5)$$

$$B = (b_1, b_2) = (3.5, 0)$$

$$C = (a_1+b_1), (a_2+b_2) = (0+3.5), (3.5+0) = 3.5, 3.5$$

The vector direction is determined by adding the X component and Y component vectors together.

The magnitude of the vector is:

$$C = (C_1, C_2) = (3.5, 3.5)$$

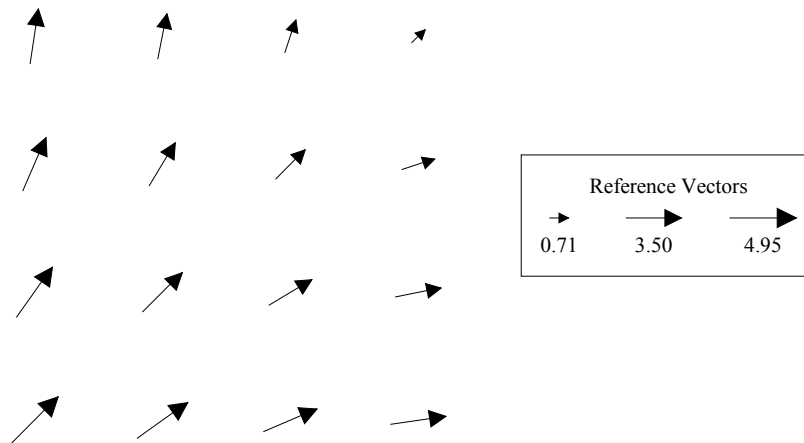
$$|C| = \sqrt{C_1^2 + C_2^2}$$

$$|C| = \sqrt{3.5^2 + 3.5^2}$$

$$|C| = \sqrt{24.5}$$

$$|C| = 4.95$$

The two grid vector components, shown on page 275, are added together to produce the map shown below.



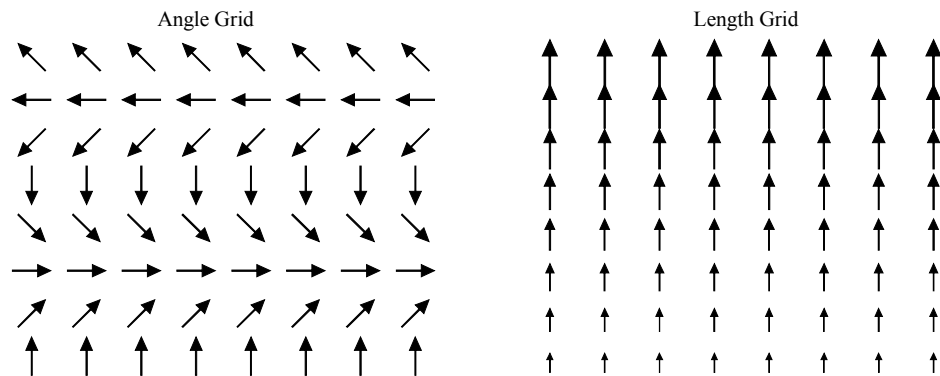
This is the resulting vector map from the X component and Y component grids shown previously.

Polar Data

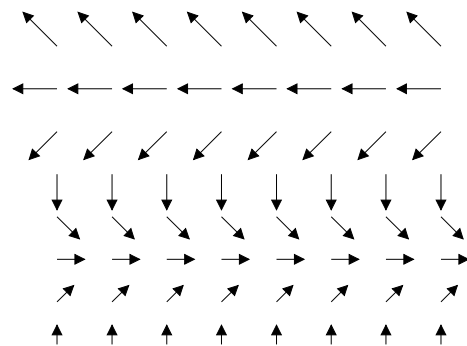
Vector maps created with polar data require that one grid contains angle (direction) data and the other contains length (magnitude) data.

Negative Data and Polar Grids

Magnitude values cannot be negative. If you have chosen a gridding technique that displays trends in the data, such as kriging, you may end up with negative values in your grid even if the original data are all positive. Regridding the magnitude data and decreasing the grid spacing is one way to make the grid honor the data better. If you regrid the magnitude data you must also regrid the angle data using the same grid line geometry. If you do not want to change the grid line geometry of the grids, or if decreasing the grid lines does not eliminate the negative data, then you can use **Grid | Math** to set all negative values equal to zero. Please see *Chapter 18, Grid Operations*, for more information on eliminating negative grid values with **Grid | Math**. Or, use a gridding method that does not trend the data such as triangulation with linear interpolation or natural neighbor.




The example grids above show the angle grid on the left and the length grid on the right. These two grids are combined to result in a vector map shown below:



This is a two-grid vector map created with polar data (angle grid and length grid).

Creating a New Vector Map

To create a 1-grid vector map:

1. Click **Map | Vector Map | New 1-Grid Vector Map** or click the  button.
2. Select a grid file in the **Open Grid** dialog and click the *Open* button.
3. The map is created.

To create a 2-grid vector map:

1. Click **Map | Vector Map | New 2-Grid Vector Map**.
2. Select a grid file in the **Open Grid** dialog. You are prompted to choose a grid file two times when creating a 2-grid vector map. The first grid file is the X component grid file or angle component grid file, and the second grid file is the Y component grid file or the length component grid file. The difference between the two files depends on the type of data you are mapping (Cartesian or polar).
3. The map is created.

Editing Vector Map Properties



To change the features of a vector map, open the vector map properties dialog by:

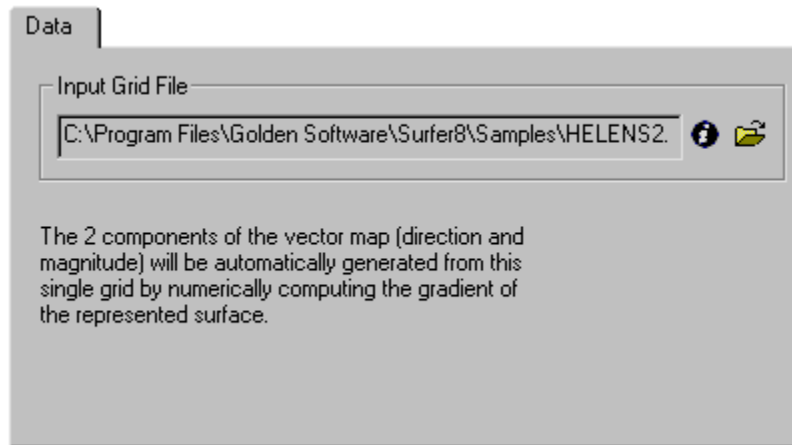
- double-clicking on the map in the plot window,
- double-clicking on the map name in the **Object Manager**,
- right-clicking the map and choosing **Properties** in the context menu,
- or by clicking **Edit | Properties** when the vector map is selected.

Vector Map Properties

The vector map properties dialog contains seven pages: **Data**, **Symbol**, **Scaling**, **View**, **Scale**, **Limits**, and **Background**. The **Data** page differs depending on whether you are editing a 1-grid vector map or a 2-grid vector map. **View**, **Scale**, **Limits**, and **Background** are discussed in *Chapter 14, Common Map Properties*.

Data - 1-Grid Vector Map

The **Data** page contains an *Input Grid File* field. You can change the grid file in the *Input Grid File* field by clicking the  button to the right of the file name. Click the  button for information about the grid. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.





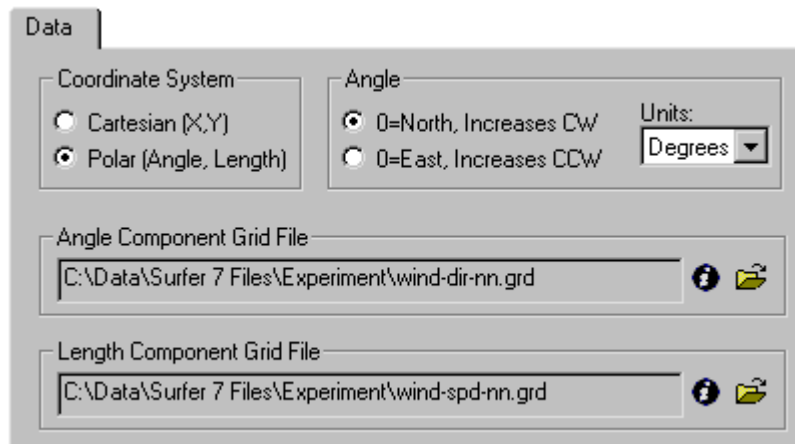
*You can obtain information about the grid file or change the grid file on the 1-grid vector map **Data** page.*

Data - 2-Grid Vector Map

The 2-grid vector map **Data** page contains the following options:

Grid Files

You can change the grid files listed on the **Data** page by clicking the  button to the right of the file name. Click the  button for information about the grid. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.



*You can set the coordinate system for the 2-grid vector map on the **Data** page.*

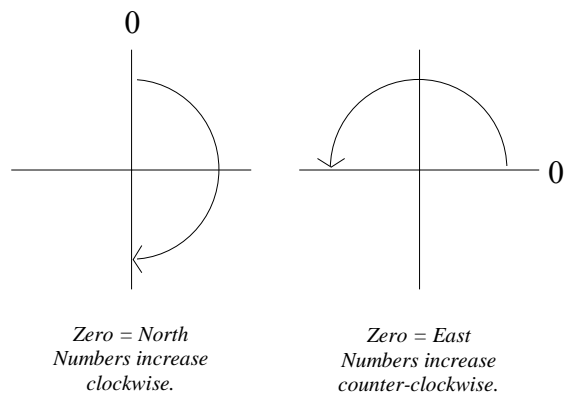
Coordinate System

Cartesian data or polar data can be used in a 2-grid vector map. The *Cartesian* coordinate option requires one grid file with X magnitude components and the second grid file with Y magnitude components. The *Polar* coordinate option requires one grid with angle (direction) information and the second grid with length (magnitude) direction for the vectors. The *Length Component Grid File* cannot contain negative values.

Chapter 11 - Vector Maps

Angle

When working with polar coordinates, set the location of the zero value to the north or east in the *Angle* group.



You can use degree or radian *Units* in a polar coordinate two-grid vector map.

Symbol

The vector map **Symbol** page contains the following options:

Style

The *Style* list contains several symbols to use to display the vectors. Click on a symbol to select it for your map.

Clipping

Check the *Clip Symbols at Map Limits* check box to truncate the symbols at the edges of the map. The symbols may extend beyond the map axes if this check box is not checked.


Properties

The *Properties* group contains buttons to change the symbol outline (style, color, and width) and fill (color and pattern). If you enable the options in the *Color Scaling* group, the symbols are not outlined.

Frequency

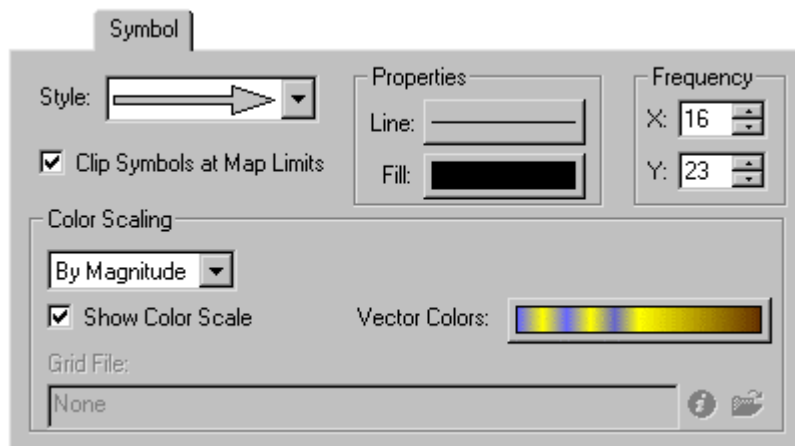
The *Frequency* settings control the number of symbols displayed on the vector map. If the frequency is set to three, every third symbol is displayed on the vector map. If there are too many symbols on the vector map, increase the frequency numbers until the map becomes legible. The *Frequency* can be set in the X and Y directions separately.

Color Scaling

The *Color Scaling* group contains options for gradationally coloring the symbols *By Magnitude* or *By Grid File*. If you select the *By Grid File* option, choose a grid file by clicking the  button to the right of the file name box. You must use a grid that contains the same grid line geometry as the vector map grid files. This means the number of rows and columns and the grid range must be the same for all grids. The vectors are colored according to the corresponding Z value in the color grid file.

Color scaling *By Magnitude* colors each vector by mapping its magnitude to a color in the specified color scale.

Click the *Vector Colors* button to select a range of colors for the vector color scale. Check the *Show Color Scale* check box to show a color scale bar for the color vector map. The check box is grayed out if the *Color Scaling* is set to *Disabled*.



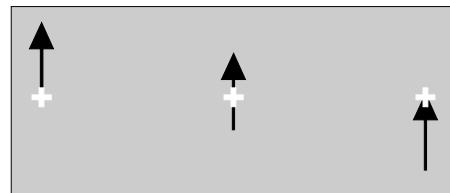
The *Symbol* page contains several options for setting symbol properties.

Scaling

The **Scaling** page contains placement and sizing options for vector symbols. Each symbol is located at a grid node.

Symbol Origin

The symbol is placed on the grid node through the *Symbol Origin* list. The symbol can be placed on the grid node at the head of the arrow symbol, at the tail of the arrow symbol, or centered on the arrow symbol.



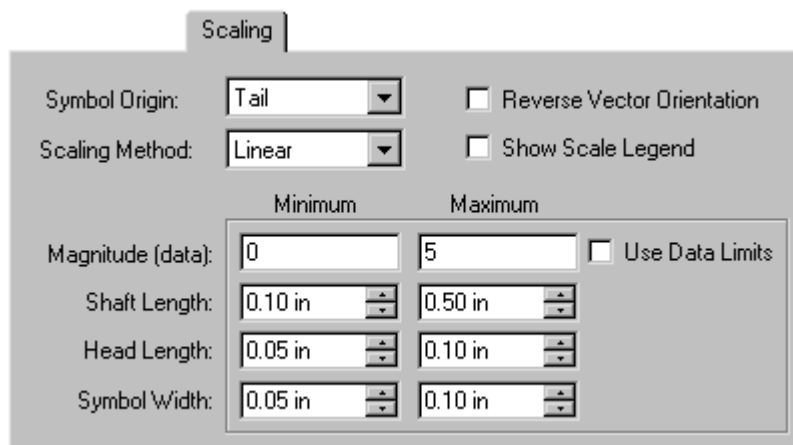
From left to right, the symbol origin at the arrow's tail, center, and head.

Scaling Method

There are three ways to scale vectors listed in the *Scaling Method* group. You can scale the symbols between the minimum and maximum data values linearly, logarithmically, or by square root. *Linear* scaling provides a better visual representation of symbols that, for the most part, are scaled in one dimension (such as arrows with varying shaft length). When scaling the arrows in two dimensions (symbol width and shaft length), *Square Root* or *Logarithmic* scaling displays the arrows more effectively. This is because the symbol size is perceived by the human eye as having an enclosed area, and area is a distance-squared function.

Orientation

The direction the arrowhead points can be reversed by clicking in the *Reverse Vector Orientation* check box.



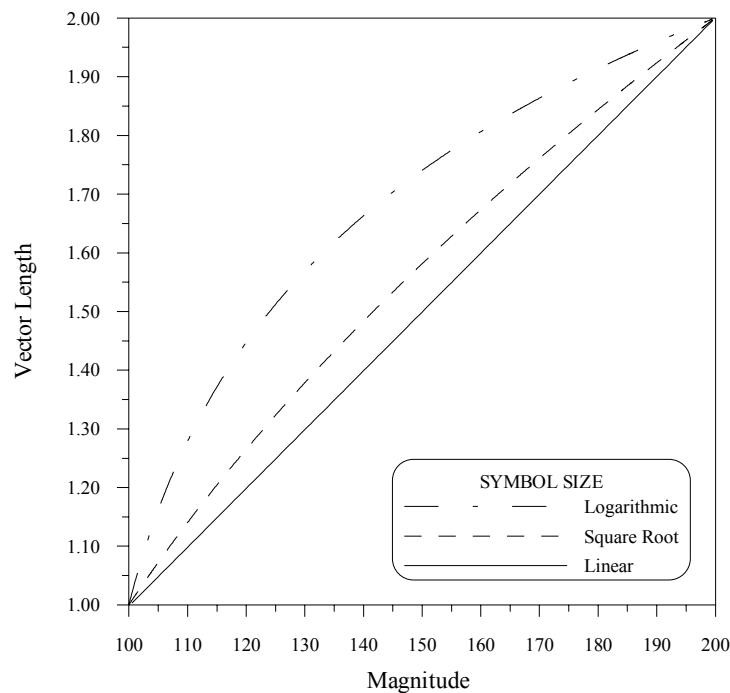
*The vector symbols scaling properties are set on the **Scaling** page.*

Legend

Check the *Show Scale Legend* check box to display a legend of the minimum and maximum vector lengths. See the *Vector Length Legend* section for more information on vector map legends.

Length and Width

You can set the minimum and maximum data values for the vectors in the *Magnitude (data)* boxes. Check the *Use Data Limits* check box to set the vector minimum and maximum to the grid minimum and maximum values. Setting a new minimum and maximum is useful in displaying a series of maps in which you would like all of the vectors to be scaled the same even though the data minimum and maximum may differ.



This graph shows the length of the vector relative to the magnitude of the vector with the three different scaling methods.

Set the range of the arrow shaft lengths, arrowhead lengths, and symbol widths in the *Shaft Length*, *Head Length*, and *Symbol Width* boxes by entering a new value into the boxes.

Color Scale Bars

A color scale bar is a legend that shows the colors used in a vector map.

Adding a Color Scale Bar

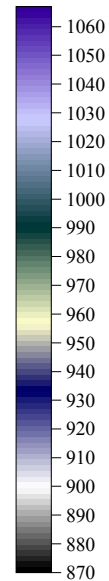
To add a color scale bar to a vector map:

1. Open the map properties dialog using one of the methods listed in the *Editing Vector Map Properties* section.
2. Select fill properties for the vectors in the *Color Scaling* group on the **Symbol** page.
3. Check the *Show Color Scale* box to activate the color scale bar.
4. Click OK or *Apply* to draw the map with a color scale bar.

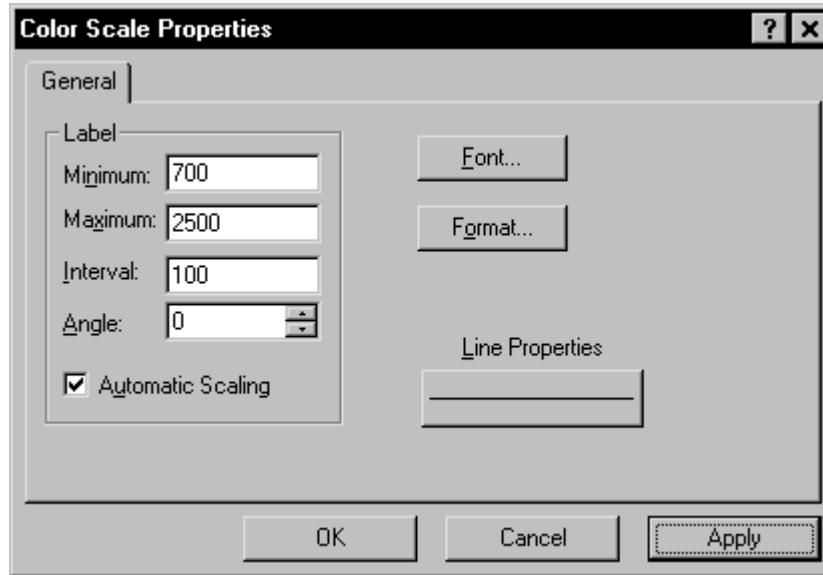
Editing a Color Scale Bar

To modify the color scale bar:

1. Double-click the color scale bar to display the **Color Scale Properties** dialog.
 - Set the first label value in the *Minimum* box.
 - Set the last label value in the *Maximum* box.
 - Define the spacing between labels in the *Interval* box.
 - Choose the angle at which the labels are drawn in the *Angle* box. Positive values rotate the label counterclockwise.
 - The *Automatic Scaling* check box sets the label frequency so labels do not overlap. If there are too many labels to fit into the space next to the color scale bar, some labels are dropped to make the labels legible.
 - Click the *Font* button to set the label text properties.
 - Click the *Format* button to choose the label numeric format (i.e. number of decimal places).
 - The *Line Properties* button allows you to specify the line properties for the color scale bar border.
2. Once the changes are made, click OK or *Apply* to redraw the color scale bar.



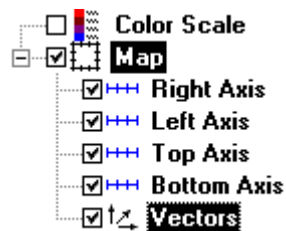
Color scale bars are added through the vector map properties dialog.



The **Color Scale Properties** dialog controls the display of the color scale bar.

Color Scale Bar Visibility

The color scale bar can be toggled on and off through the **Object Manager**.



Remove the check mark from the Color Scale box in the **Object Manager** to make the color scale bar invisible.

Removing a Color Scale Bar

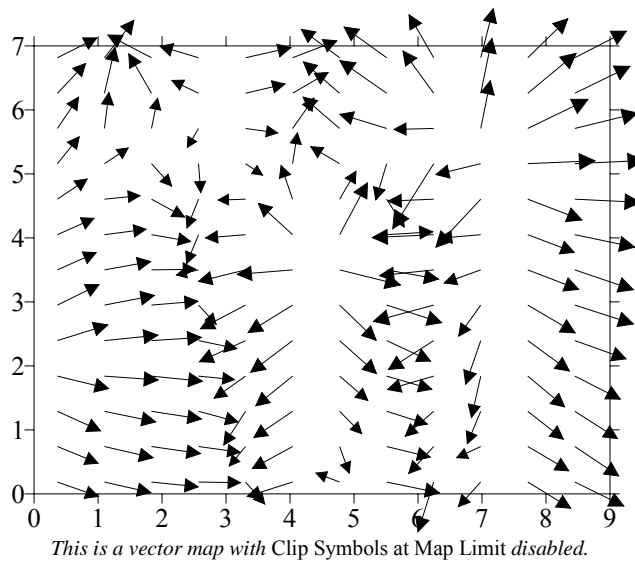
A color scale bar can be removed by clicking on the scale bar and pressing the DELETE key on your keyboard. Alternatively, you can uncheck the *Show Color Scale* check box in the map properties dialog.

Clipping Symbols on Vector Maps

The **Symbol** page in the vector map properties dialog contains a *Clip Symbols at Map Limits* check box. This option allows you to control the display of the symbols at the edges of your map. If the *Clip Symbols at Map Limits* box is checked, any symbols at the edge of the map are truncated at the map limits. If the *Clip Symbols at Map Limits* box is not checked, the entire symbol is drawn.

Clipping and Map Limits

If the map limits are decreased via the map properties **Limits** page, no part of the symbol is drawn if the symbol's grid node falls outside of the map limits. You can expand the map limits or change the axes properties to display the entire symbol without overlapping the map axes.

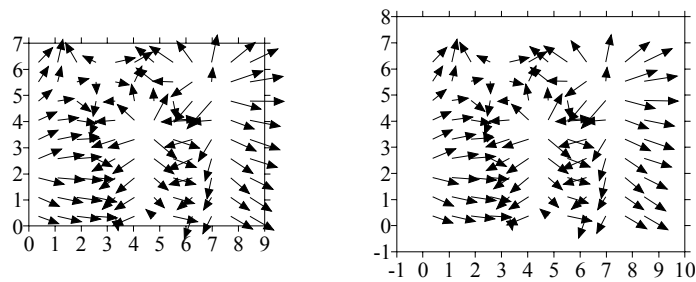


Changing the map limits:

1. Create a vector map.
2. Open the map properties using one of the methods listed in the *Editing Vector Map Properties* section.
3. On the **Symbol** page, make sure that the *Clip Symbols at Map Limits* option is not checked.
4. Select the **Limits** page.
5. Decrease the values in the *xMin* and *yMin* boxes.
6. Increase the value in the *xMax* and *yMax* boxes.

Alternatively, you could change the properties of all four axes. See *Chapter 15, Axes* for more information on axes.

1. Double-click on an axis in the plot window or in the **Object Manager** to open the axis properties.
2. On the **Scaling** page in the axis properties dialog, change the numbers in the *Axis Minimum*, *Axis Maximum*, *First Major Tick*, *Last Major Tick*, and *Cross X (or Y) Axis at* fields.



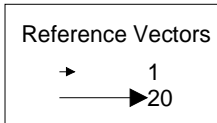
The map on the left displays the vectors without Clip Symbols at Map Limits checked. To avoid overlapping the axis with the symbols, you can expand the map limits or edit the axis properties.

Vector Length Legend

The **Scaling** page in the vector map properties dialog contains a *Show Scale Legend* check box that is used to turn on or off the legend. The legend displays the minimum and maximum vector length as symbols along with a numeric value for each symbol. If a color scale was used on the **Symbol** page (*By Magnitude* or *By Grid File*), the symbols in the legend are displayed with the minimum and maximum colors. Initially, the legend appears with a "Reference Vectors" title along with the minimum and maximum vector lengths as set in the *Magnitude (data)* group on the **Scaling** page.

Editing the Vector Length Legend

To edit the legend, double-click on the legend or double-click on the word *Legend* in the **Object Manager**.

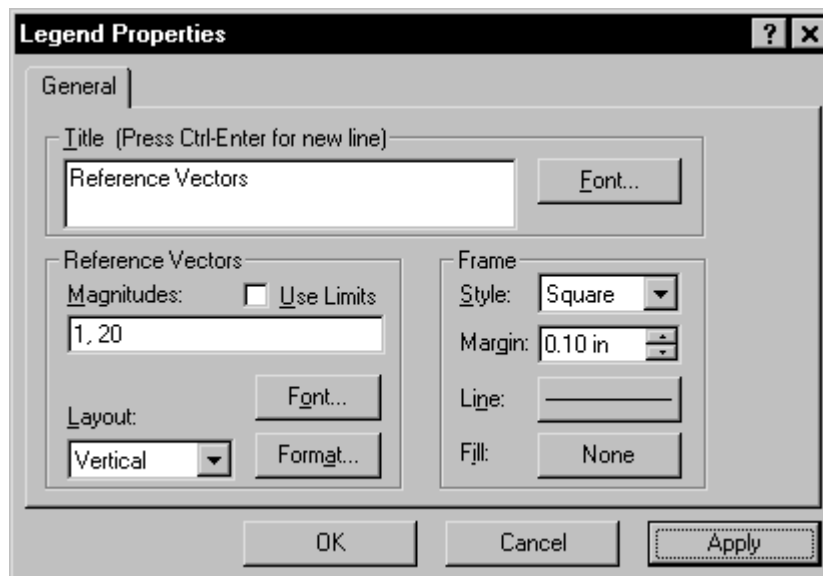


This is an example of a vector length legend created with the settings shown in the dialog below.

Legend Properties Dialog

Title

To enter a new title for the legend, highlight the default title, "Reference Vectors," and then enter a new title. You can click the *Font* button to change the title face, size, style, and color.



*You can open the **Legend Properties** dialog by double-clicking on the legend in the plot window or in the **Object Manager**.*

Reference Vectors

If the *Use Limits* check box is checked in the *Reference Vectors* group, the length of the vectors are determined by the values entered in the *Magnitude (data)* boxes on the **Scaling** page. If you would like to display different vector lengths in the legend, enter the new lengths in the *Magnitudes* box in the **Legend Properties** dialog. The new vectors are scaled relative to the scaling set on the **Scaling** page in the vector map properties dialog. You can display more than two vectors in the legend by entering a string of numbers separated by commas or spaces into the *Magnitudes* box.

The *Magnitudes* box in the dialog pictured on the previous page shows two vectors drawn in the legend, one for the minimum 1 length and one for the maximum 20 length. To add additional vectors in the legend, enter numbers separated by commas or spaces into the *Magnitudes* box. For example, 1, 5, 15, 20 would show four symbols in the vector legend.

You can align the symbols vertically or horizontally in the legend by selecting *Vertical* or *Horizontal* in the *Layout* list. Click the *Font* button in the *Reference Vectors* group to change the symbol label face, size, style, and color. Click the *Format* button to change the numeric format of the labels. For example, the number of decimal places on the numbers is changed by clicking the *Format* button and making the necessary changes in the **Label Format** dialog.

Frame

You can set the frame around the legend to *Square*, *Rounded*, or *None*. Choose *None* to display the legend without a frame. The *Margin* is the distance between the legend text and the frame. Set the margin to a number between zero and four inches. The frame line style, color, and width can be changed by clicking the *Line* button and the background fill pattern and color can be changed by clicking the *Fill* button.

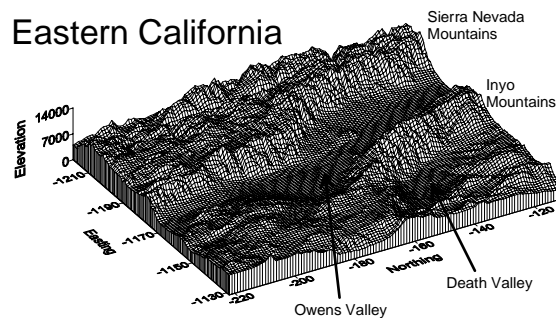
Chapter 12

Wireframes

Introduction to Wireframes

Wireframes are three-dimensional representations of a grid file. Wireframes are created by connecting Z values along lines of constant X, Y, and Z. At each XY intersection (grid node), the height of the wireframe is proportional to the Z value of that node. The number of columns and rows in the grid file determines the number of X and Y lines drawn on the wireframe.

Eastern California



Wireframes are three-dimensional representations of a grid file.

Wireframes can display any combination of X lines, Y lines, or Z contour lines. On the wireframe, X lines correspond to the columns in the grid file and Y lines correspond to rows in the grid file. Lines of equal Z value on the wireframe are drawn at the appropriate elevation on the surface.

Wireframes can display color zones assigned to different Z levels. The plotted X, Y, or Z lines are displayed in the color assigned to the specific levels. Color zones can be assigned to any combination of X, Y, and Z lines.

Wireframe Limits

The grid limits define the extent of wireframes. When creating a grid file, you can set the grid limits to the desired limits for the wireframe. The wireframe cannot be larger or smaller than the extent of the grid file. Use the **Grid | Extract** command to produce a subset of a grid file; this is especially useful in thinning a grid file. Use **Grid | Spline Smooth** to increase the density of a grid file in order to add additional X and Y lines to the wireframe. Use the **Grid | Data** command to create a new grid file with different limits, or use **Grid | Mosaic** to merge multiple grid files. See *Chapter 4, Creating Grid Files* and *Chapter 18, Grid Operations* for more information on modifying grid files.

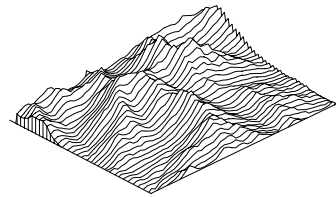
Wireframe Grid Files

The grid file is read only once when the wireframe is first created. If you change the grid file used to produce the wireframe, the grid file changes are not reflected in the wireframe. You can even delete the grid file and the wireframe is still drawn as before. Recreate the wireframe to use an updated grid file.

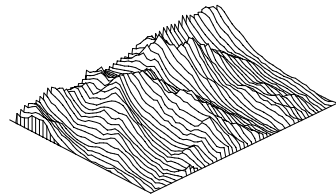
Wireframe Components

Wireframe components include the following:

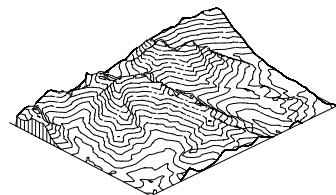
- X lines represent the columns in the grid file. The number of X lines on a wireframe is controlled by the number of columns in the grid file.
- Y lines represent the rows in the grid file. The number of Y lines on a wireframe is controlled by the number of rows in the grid file.
- Z lines are contour lines drawn on the wireframe. The number of Z lines on a wireframe depends on the minimum and maximum contour levels and the specified contour interval.
- Color zones can be assigned to different Z levels on wireframes. **Surfer** can create a spectrum of colors between two specified colors for the minimum and maximum levels of the wireframe; alternatively, you can specify colors for particular levels.
- The base is the block on the bottom of the wireframe. It can extend to any Z level. The sides of the base can be highlighted by adding vertical base lines, which are the extensions of the X and Y lines from the wireframe to the bottom of the base.
- Map scale controls the size of the map and the number of map units per page unit.



This is a wireframe with lines of constant Y.



This a wireframe using the same grid file as above, with lines of constant X.




This is a wireframe using the same grid file as above, with lines of constant Z.

- Map axes display tick marks and axis labels along the borders of the map. Axes show the X, Y, and Z scaling in map units. You can include grid lines as extensions of the tick marks over the extent of the map.
- Map orientation is defined as the rotation, tilt, and field of view of the map.
- A background can be added to a map. On wireframes, the map background covers the bottom plane of the base.

Creating a New Wireframe

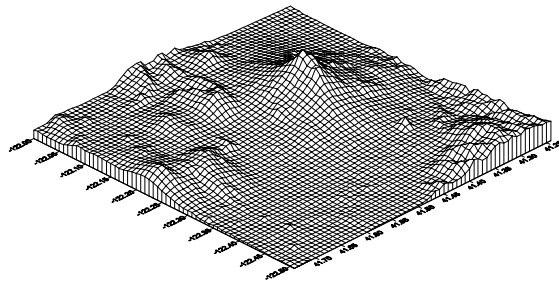
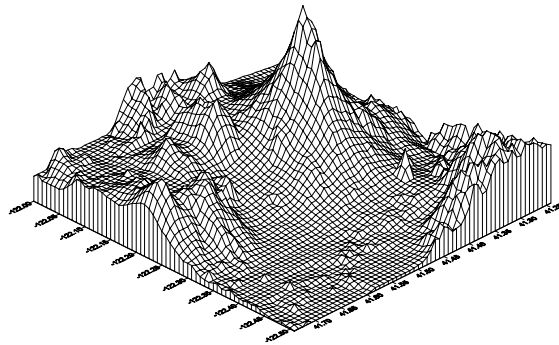
To create a new wireframe:

1. Click **Map | Wireframe** or click the  button.
2. Select a grid file in the **Open Grid** dialog and then click *Open*.
3. The map is created.

Editing Wireframe Properties

To change the features of the wireframe, open the wireframe properties dialog by:

- double-clicking on the wireframe in the plot window,
- double-clicking on the wireframe name in the **Object Manager**,
- right-clicking the wireframe and choosing **Properties** in the context menu,
- or by clicking **Edit | Properties** when the wireframe is selected.



*These are two wireframes of the same grid, each using a different Z scale. The Z scale is exaggerated in the top view. Use the **Scale** page to change the length of the Z axis.*

Wireframe Properties


The wireframe properties dialog contains six pages: **General**, **Z Levels**, **Color Zones**, **View**, **Scale**, and **Background**. **View**, **Scale**, and **Background** are discussed in *Chapter 14, Common Map Properties*.

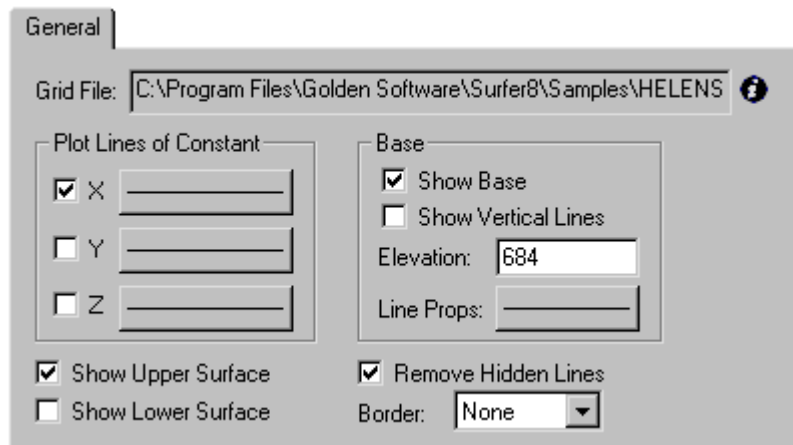
General

The wireframe properties **General** page contains the following options:

Grid File

The file used to create the wireframe is listed in the *Grid File* group. The grid file cannot be changed for existing wireframes.

Click the  button for information about the grid. The information includes the number of rows and columns in the grid, statistics, and the minimum and maximum X, Y, and Z values contained in the file. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.



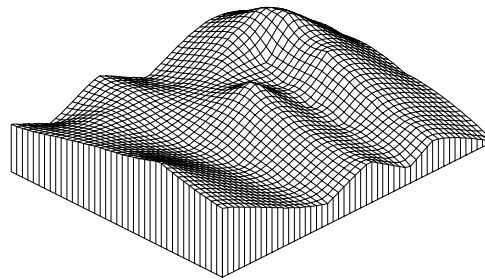
*The **General** page in the wireframe properties dialog controls which lines are displayed and it controls the base properties.*

Plot Lines of Constant

The *Plot Lines of Constant* group allows you to specify the lines to be used when displaying the wireframe. Any combination of X, Y, and Z lines may be used.

Base

The *Base* group controls the display of a base on the wireframe. The base is the area "underneath" the wireframe. The bottom of the base can be drawn at any Z level relative to the surface. Check the *Show Base* box to display the base. You can adjust the elevation of the bottom of the base in the *Elevation* box.



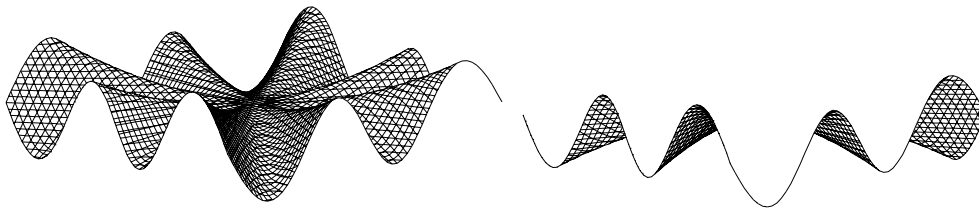
This is a wireframe with a base. Vertical base lines are included to emphasize the base.

Add X and Y lines to the base by checking the *Show Vertical Lines* check box. You can change the vertical line color, style, and width by clicking on the *Line Props* button.

Show Surface

The *Show Upper Surface* check box displays only the top portion of the surface. If the *Remove Hidden Lines* check box is activated, only those portions of the upper surface not obscured by other portions of the surface are drawn.

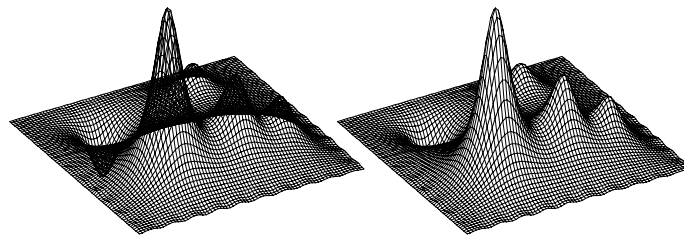
The *Show Lower Surface* check box displays only the bottom of the surface. If the *Remove Hidden Lines* check box is activated, only the places where the bottom of the wireframe is not obscured by other portions of the surface are drawn. A base cannot be displayed when viewing the lower surface.



The same wireframe is displayed in both maps. Only the upper surface is displayed on the wireframe on the left. Only the lower surface is displayed on the right. Both maps have Remove Hidden Lines activated.

Remove Hidden Lines

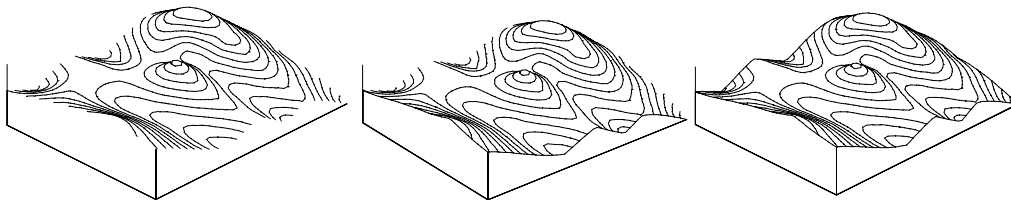
The *Remove Hidden Lines* check box removes the display of X and Y lines behind other lines on the wireframe. When hidden lines are not removed, the wireframe takes on a transparent appearance.



The wireframe on the left displays hidden lines. The wireframe on the right has hidden lines removed.

Border

The *Border* list is used to display a line representing the intersection of the wireframe with the sides of the base. You can display the border on the *Front Only*, on *All Sides*, or on *None* of the sides. The border line is not visible on wireframes displaying both X and Y lines.



The wireframe on the left does not display border lines. The middle wireframe displays borderlines on the front only. The right wireframe shows borderlines on all sides, adding definition to the edge of the wireframe.

Z Levels

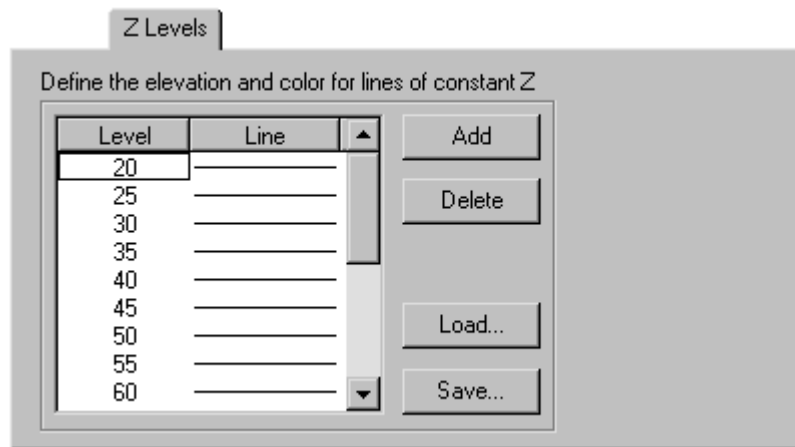
The **Z Levels** page controls the display of lines of constant Z. When displaying the Z lines on a wireframe, use the options on the **Z Levels** page to control which levels of Z lines to display. You can specify the Z contour levels and the line properties for the individual Z lines on this page. The selected levels affect the Z contours on the wireframe and do not display contour labels or hachures.

Levels

Evenly spaced Z levels are assigned by clicking the *Level* button. This displays the **Contour Levels** dialog, where you can set the minimum and maximum contour levels as well as the contour interval.

You can also double click on a level number change the value. Unevenly spaced levels are created in this manner.

Click the *Add* button to add a level midway between the selected level and the next lower level or highlight a level and then click the *Delete* button to remove a selected level. Adding and deleting levels creates irregularly spaced contours.



The Z Levels page is used to control the properties of lines of constant Z.

Lines

Click the *Line* button to create gradational line color. Alternatively, you can double-click on a single line to change individual line properties.

Level Files

If you have previously created a level file [.LVL], click the *Load* button to load the level file. If you would like to use a newly defined level file in other wireframes, click the *Save* button to save it as a level file. Level files are discussed in *Appendix C, File Formats*.

Color Zones

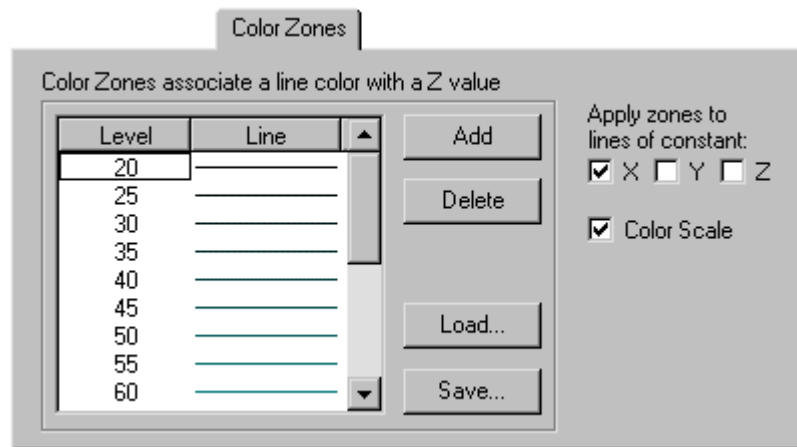
The wireframe **Color Zones** page controls the color to apply to lines of constant X, Y, and Z based on Z values in the grid file.

Levels

Evenly spaced Z levels are assigned by clicking the *Level* button. This displays the **Contour Levels** dialog, where you can set the minimum and maximum contour levels as well as the contour interval.

You can also double click on a level number change the value. Unevenly spaced levels are created in this manner.

Click the *Add* button to add a level midway between the selected level and the next lower level or highlight a level and then click the *Delete* button to remove a selected level. Adding and deleting levels creates irregularly spaced contours.



*The **Color Zones** page controls the line properties of lines of constant X, Y, and Z.*

Lines

Click the *Line* button to create gradational line color. Alternatively, you can double-click on a single line to change individual line properties.

Check the lines to apply the color to in the *Apply zones to lines of constant* X, Y, and Z check boxes.

Level Files

If you have previously created a level file [.LVL], click the *Load* button to load the level file. If you would like to use a newly defined level file in other wireframes, click the *Save* button to save it as a level file. Level files are discussed in *Appendix C, File Formats*.

Color Scale Bar

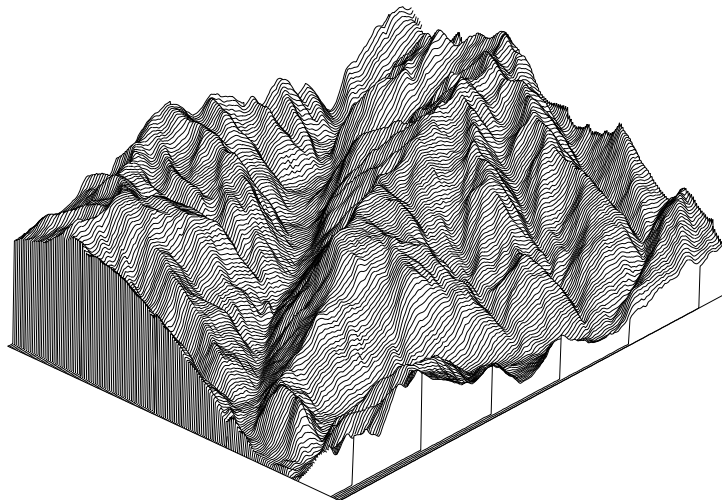
A color scale bar can be displayed on the map by checking the *Color Scale* check box on this page.

Specifying Lines

Any combination of X, Y, and Z lines can be displayed on wireframes. When using X and Y lines to represent the surface, a mesh appears on top of the surface. When using the Z lines, an elevated contour map is drawn.

To specify which lines to draw on the wireframe:

1. Open the wireframe properties using one of the methods in the *Editing Wireframe Properties* section.
2. In the *Plot Lines of Constant* group on the **General** page, click the appropriate check boxes to display any combination of X, Y, and Z lines on the wireframe.



This is a wireframe with only Y lines displayed.

Chapter 12 - Wireframes

When displaying the Z lines on a wireframe, use the options on the **Z Levels** page to control which levels of Z lines to display. You can set the Z contour levels and the line properties for the individual Z lines on this page. Wireframe contours do not display contour labels or hachures.

Note: To display contour labels or hachures on a wireframe, create a contour map using the **Map | Contour Map | New Contour Map** command. Overlay the contour map on the wireframe using the **Map | Overlay Maps** command.

3. Set any other desired options on the **General** page.
4. Click OK or *Apply* and the wireframe is displayed with the line types you have chosen.

Line Properties

Line properties for wireframes are controlled from the wireframe properties dialog. The X and Y line properties specified in the *Plot Lines of Constant* group on the **General** page are applied over the entire surface without regard to the relative level of the surface. To assign X and Y line properties so that they change over the vertical range of the wireframe, set the line properties on the **Color Zones** page. Color zone line properties vary based on Z values in the grid file. See the *Color Zones* section in this chapter for more information on using the properties in the **Color Zones** page.

Use the line button in the *Plot Lines of Constant* group on the **General** page to apply the same line properties to all lines of constant Z without regard to level. To apply Z line properties that vary according to elevation, use the options in the **Z Levels** or **Color Zones** page. See the sections *Z Line Levels and Properties* and *Color Zones* in this chapter for more information.

Line Property Precedence

Wireframe line properties can be assigned three different ways depending on the effect you are trying to achieve:

1. On the **Color Zones** page,
2. On the **Z Levels** page, and
3. On the **General** page.

The highest precedence is applied to line properties assigned on the **Color Zones** page. When the *Apply zones to lines of constant* check boxes on the **Color Zones** page are enabled, those properties take precedence over any other assigned line properties.

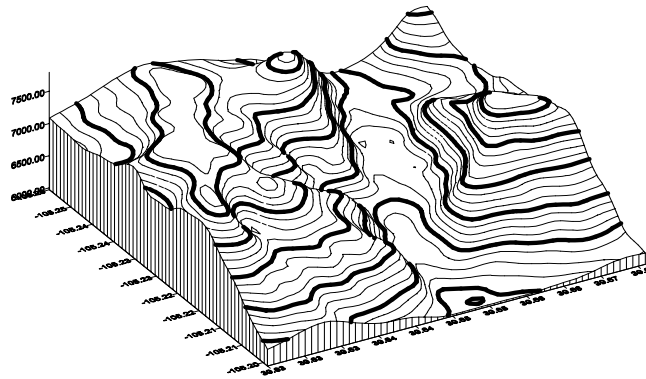
Line properties assigned in the *Plot Lines of Constant* group on the **General** page have the same precedence as the line properties assigned on the **Z Levels** page. If the options on the **Color Zones** page are not used, the next highest precedence are selections on the **General** page or **Z Levels** page, whichever is changed last. (The **Z Levels** page colors are only used on Z lines.)

Z Line Levels and Properties

The levels of lines of constant Z are controlled on the **Z Levels** page of the wireframe properties dialog. In addition to specifying levels, the line properties for individual levels can also be selected. The line properties assigned in the **Z Levels** page are applied only to Z lines when *Apply zones to lines of constant Z* on the **Color Zones** page is not activated.

Using both the **Z Levels** options and the **Color Zones** options allows you to create wireframes with varying color combinations. If you want the X and Y lines displayed with various colors based on the grid node values, use the **Color Zones** options. By not applying the color zones to lines of constant Z, the properties specified in the **Z Levels** page are used instead.

As an example, consider a wireframe with Z values ranging from 0 to 100 feet. On the **Z Levels** page, contour lines are displayed every 4 feet. In the **Color Zones** page, the X and Y line properties are specified to change every 50 feet. When the wireframe is drawn, Z lines are drawn every four feet and the X and Y properties change every 50 feet.

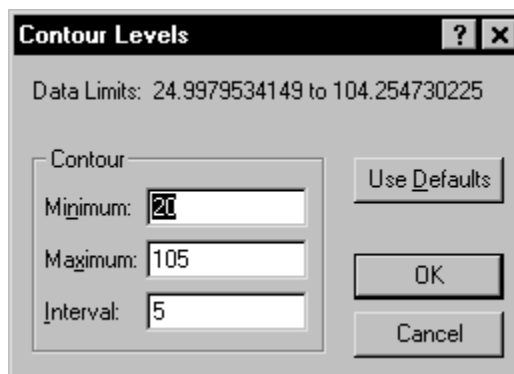


You can assign color zones based on different Z levels than the contour lines plotted on the wireframe.

Specifying Evenly Spaced Z Levels

To create evenly spaced Z levels:

1. Open the wireframe properties using one of the methods listed in the *Editing Wireframe Properties* section.
2. Click the **Z Levels** page.
3. Click the *Level* button to open the **Contour Levels** dialog.
4. Enter the values for the *Minimum* and *Maximum* Z levels (contours) and the *Interval* value. The *Interval* defines the spacing in Z units between adjacent contour lines.
5. Click OK and the level list is updated.

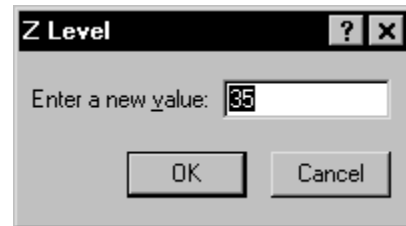


*Open the **Contour Levels** dialog by clicking on the *Level* button on the **Z Levels** page.*

Adding, Deleting, or Modifying Z Level Values

To add, delete, or modify Z levels:

1. Open the wireframe properties using one of the methods listed in the *Editing Wireframe Properties* section.
2. Click the **Z Levels** page.
3. Click the *Add* button to add a contour level midway between the selected level and the next lower level.
4. Use the *Delete* button to remove the selected level from the list.
5. To modify an existing level, double-click the level value. Type the new level value into the **Z Level** dialog. Click OK and the level list is updated.

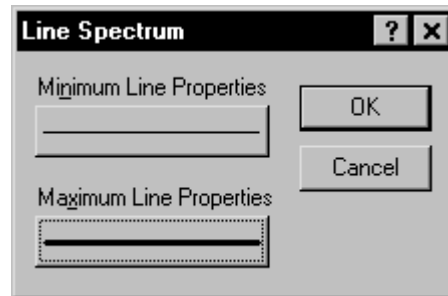


*You can enter a new value for the contour level in the **Z Level** dialog.*

Assigning Z Level Properties Based on a Line Spectrum

To assign Z level line properties:

1. Open the wireframe properties using one of the methods listed in the *Editing Wireframe Properties* section.
2. Click the **Z Levels** page.
3. Click the *Line* button to display the **Line Spectrum** dialog.
4. Click the *Minimum Line Properties* and *Maximum Line Properties* buttons to specify the line style, color, and width for the minimum and maximum Z levels.
5. Click OK and the **Z Levels** page line properties are updated.

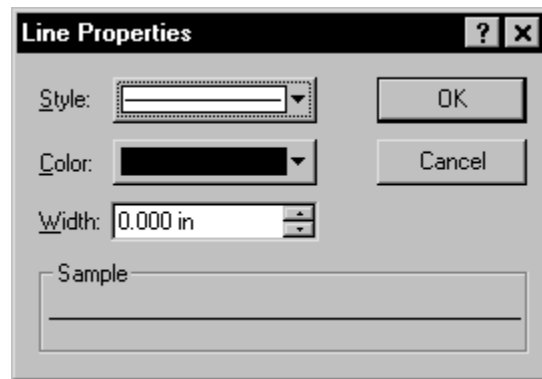


*The **Line Spectrum** dialog is opened by clicking the *Line* button on the **Z Levels** page.*

Assigning Properties to Individual Z Levels

To assign Z level line properties to individual levels:

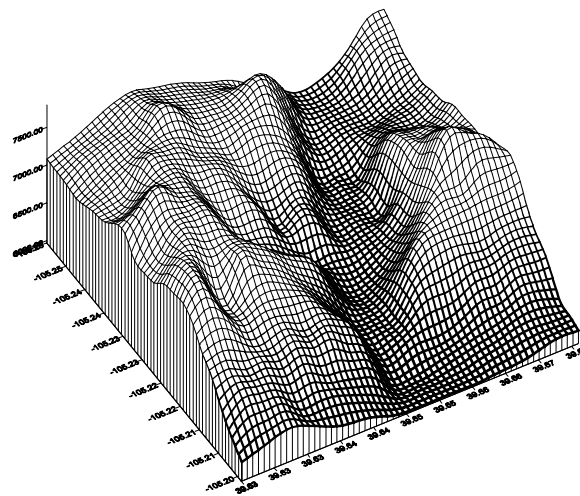
1. Open the wireframe properties using one of the methods listed in the *Editing Wireframe Properties* section.
2. Click the **Z Levels** page.
3. You can assign a specific line style, color, or width to a specific contour level by double-clicking the line sample displayed for that level. Select the line properties from the **Line Properties** dialog. Assign properties in this manner for any of the levels.
4. Click OK and the **Z Levels** page is updated to reflect the changes.



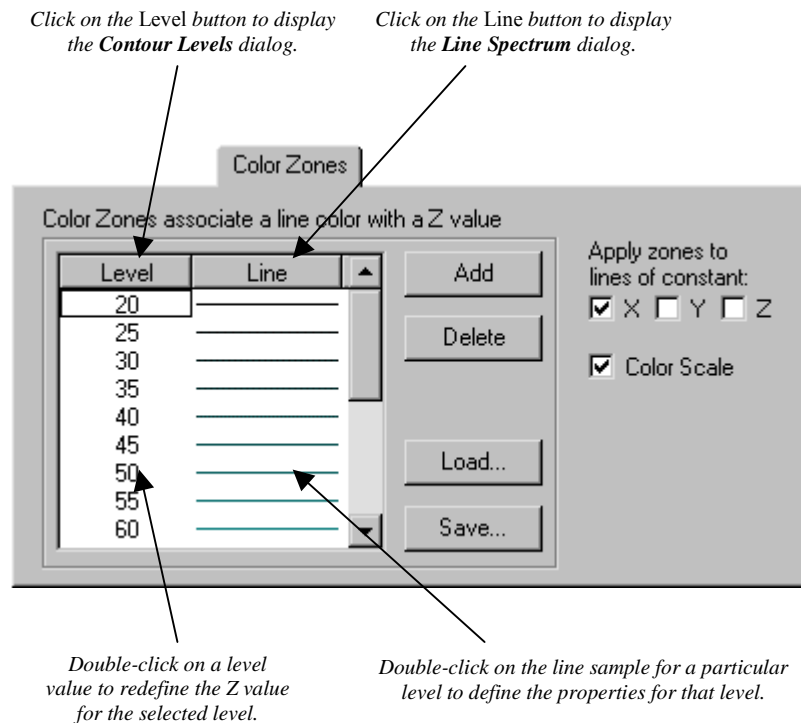
*Open the **Line Properties** dialog by double clicking on the line sample on the **Z Levels** page.*

Color Zones

Color zones refer to the colors assigned to the X, Y, and Z lines on a wireframe that vary according to the Z value in the grid file. Color zones can be assigned automatically to the different levels or they can be assigned specifically to an individual level.



*A gradual change in both colors and line width can be assigned on the **Color Zones** page. A gradual change in line width is shown on this plot.*



Specifying Color Zones

To set the color zones:

1. Open the wireframe properties using one of the methods listed in the *Editing Wireframe Properties* section.
2. Click the **Color Zones** page.
3. You can assign evenly spaced Z levels by clicking the *Level* button. This displays the **Contour Levels** dialog.
4. Enter the values for the *Minimum* and *Maximum* contours as well as the *Interval* value that defines the spacing between adjacent contour lines (in Z units).
5. Click OK and the color zone list is updated.

Assigning Color Zones based on a Line Spectrum

To assign colors based on a line spectrum:

1. Open the wireframe properties using one of the methods listed in the *Editing Wireframe Properties* section.
2. Click the **General** page.
3. In the *Plot Lines of Constant* group, click in the check boxes for the lines you want to display on the wireframe. Color zones are applied only to these lines.
4. Click the **Color Zones** page. If necessary, define the Z levels by clicking on the *Level* button to set the *Minimum*, *Maximum*, and *Interval* values in the **Contour Levels** dialog.
5. Click the *Line* button to define the line color, style, and width. The line style does not change gradationally like line color and widths do.
6. Click on the *Minimum Line Properties* button in the **Line Spectrum** dialog to display the **Line Properties** dialog.
7. In the **Line Properties** dialog, specify the line *Style*, *Color*, and *Width* for the minimum value.
8. Click OK.
9. Repeat using the *Maximum Line Properties* button.
10. Click OK in the **Line Spectrum** dialog to return to the **Color Zones** page.
11. Click in the *Apply zones to lines of constant X*, *Y*, or *Z* check boxes to specify the lines to apply the color zones.
12. Click OK or *Apply* and the color zones are displayed on the wireframe.

Assigning Line Properties to Specific Color Zones

To assign line properties to specific zones:

1. Open the wireframe properties using one of the methods listed in the *Editing Wireframe Properties* section.
2. Double-click a line sample on the **Color Zones** page for the zone you want to modify to open the **Line Properties** dialog.
3. Select the line style, color, and width.
4. Click OK in the **Line Properties** dialog.
5. Repeat the procedure for all the zones you want to change.
6. Click OK or *Apply* and the map is drawn with the specified line properties.

The Z lines can be assigned on the **Color Zones** page or the **Z Levels** page. If you assign Z line color on the **Z Levels** page, be sure to uncheck the *Apply zones to lines of constant Z* check box on the **Color Zones** page. Refer to the *Z Line Levels and Properties* section of this chapter for more information on using the **Z Levels** page.

Adding and Deleting Color Zones

To add or delete color zones:

1. Open the wireframe properties using one of the methods listed in the *Editing Wireframe Properties* section.
2. Click on the **Color Zones** page.
3. Use the *Add* and *Delete* buttons to create additional color zones or to remove existing zones. The *Add* button adds a level between the selected level and the next lower level. The *Delete* button removes the selected level from the list.
4. Double-click a particular level value in the list to enter a new value for the level in the **Z Level** dialog.
5. Click OK and the color zone list is updated.

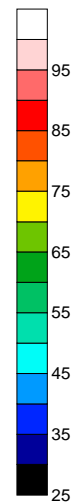
Color Scale Bar

A color scale bar is a legend that shows the fill assigned to each contour level on a wireframe. The color scale bar shows colors used in color zones.

Adding a Color Scale Bar

To add a color scale bar to a wireframe:

1. In the wireframe properties dialog, select colors on the **Color Zones** page.
2. On the **Color Zones** page in the wireframe properties dialog, check one or more of the *Apply zones to constant lines of* check boxes.
3. Check the *Color Scale* check box.
4. Click OK or *Apply* to draw the wireframe with a color scale bar.



Color Scale Bar Visibility

The color scale bar can be toggled on and off through the **Object Manager**.

Color scale bars are added through the wireframe properties dialog.

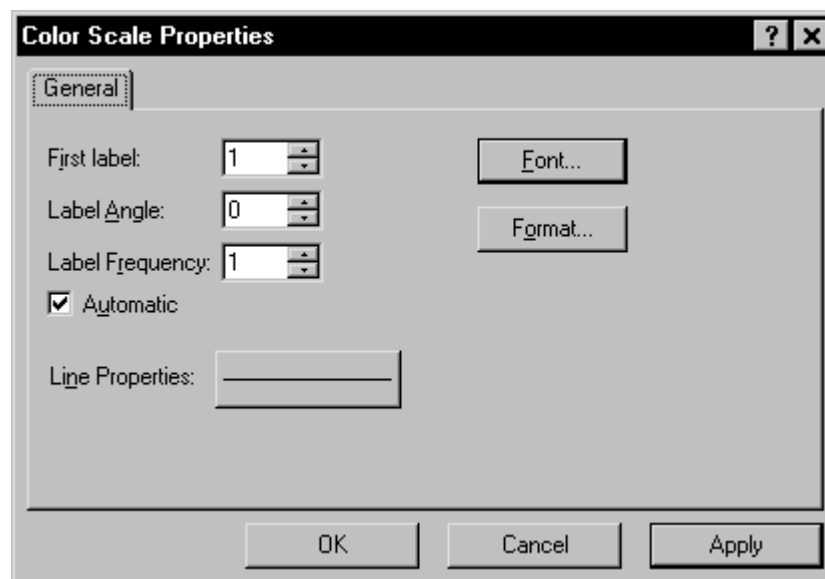
Removing a Color Scale Bar

A color scale bar can be removed by clicking on the color scale bar and pressing the DELETE key on your keyboard. Alternatively, you can uncheck the *Color Scale* check box in the map properties dialog.

Editing a Color Scale Bar

To modify a color scale bar:

1. Double-click the scale bar to display the **Color Scale Properties** dialog.
 - Indicate the level index number of the first label in the *First* label box. The level index number represents the relative position of the contour level in the ordered list (starting at one), and not the actual level value.
 - Choose the angle at which the labels are drawn in the *Label Angle* box.
 - Use the *Label Frequency* field to specify how often a label appears. A frequency of one draws every label, two draws every other label, etc.
 - The *Automatic* check box sets the label frequency so labels do not overlap. If there are too many labels to fit in the space next to the color scale bar, some labels are dropped to make the labels legible.
 - Click the *Font* button to set the label text properties.
 - Click the *Format* button to choose the label numeric format (i.e. number of decimal places).
 - The *Line Properties* button allows you to specify the line properties for the color scale bar border.
2. Once the changes are made, click OK or *Apply* to redraw the color scale bar.



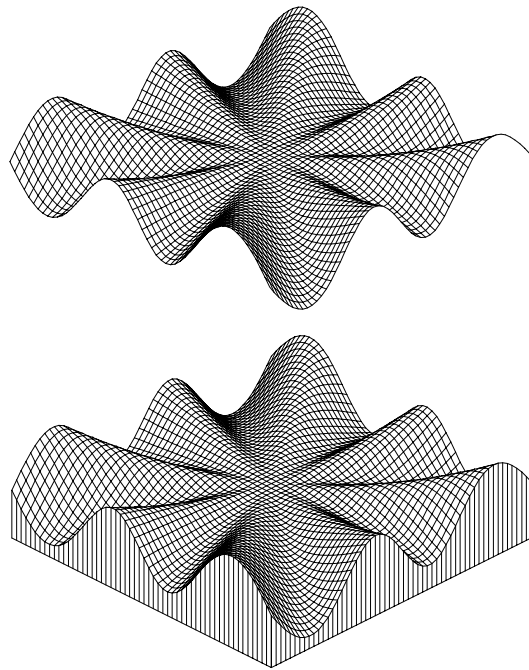
The Color Scale Properties dialog controls the display of the color scale bar.

Wireframe Base

A wireframe base is the opaque block below the wireframe. The bottom of the base can extend to any Z level. The sides of the base can be highlighted with vertical lines that are extensions of the X and Y lines.

To change the base parameters:

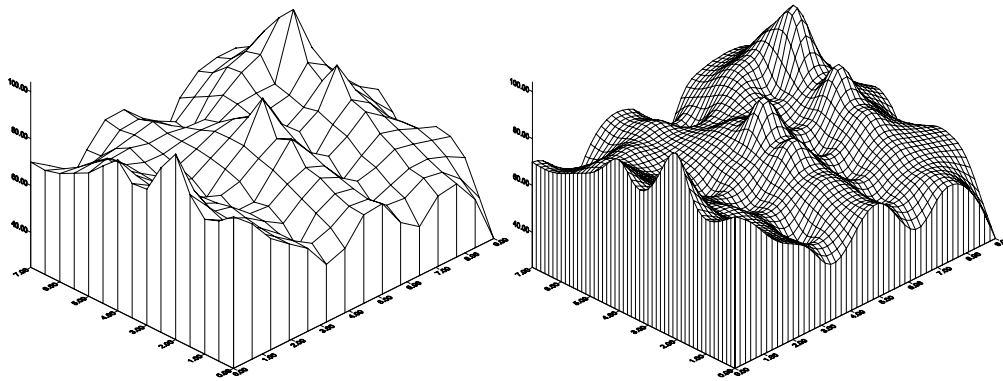
1. Open the wireframe properties using one of the methods listed in the *Editing Wireframe Properties* section.
2. The **Base** group on the **General** page controls the display of the base on the wireframe.
 - a. Activate the *Show Base* check box to display the base.
 - b. To display vertical base lines, activate the *Show Vertical Lines* check box. Vertical lines are drawn on the block face between the base and the wireframe border. The vertical lines represent the X and Y grid lines extended from the border of the wireframe to the base.
 - c. To specify base height, enter the value into the *Elevation* box. The value is in Z data units. The smaller the number entered in the *Elevation* box, the farther the base begins from the bottom of the wireframe surface, creating a thicker base.
 - d. Specify the line properties for the base by clicking the *Line Props* button and displaying the **Line Properties** dialog.
3. Click **OK** or *Apply* and the wireframe is drawn with the specified base parameters.



The wireframe on top does not display a base or axes. The wireframe on the bottom displays a base with vertical base lines.

Smoothing a Wireframe

The smoothness of wireframes is a function of the density and topography of the grid. For example, a 10 by 10 grid file (ten rows and ten columns) appears much more angular than a 50 by 50 grid file created from the same data.

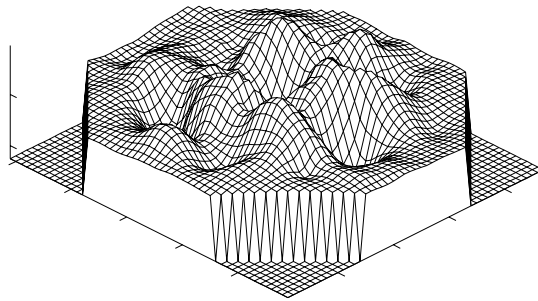


A wireframe using a higher grid density appears much smoother than a wireframe at a lower grid density. The wireframe on the left is from a 15 x 15 grid. The wireframe on the right is from a 50 x 50 grid.

You can increase the number of grid rows or columns in one of two ways. One method is to create a new grid file with more lines in the X and Y directions in the **Grid Data** dialog. Alternatively, use the **Grid | Spline Smooth** command to insert additional rows and columns into an existing grid file. Also, the **Grid | Filter** command can be used to increase the smoothness of the map.

Blanking on a Wireframe

Blanked regions of a grid file are represented by low flat areas on a wireframe. The level of the blanked region is set to the minimum Z value for the grid file. For example, consider a blanked grid file that ranges from zero to 100 in the Z dimension. When a wireframe is displayed, the blanked region appears as a large flat area at the Z=0 elevation. See *Chapter 18, Grid Operations* for more information on blanking.



Blanked regions on a wireframe are drawn as low, flat areas.

Digital Elevation Models (DEM) may have some blanking along the edges.

The DEM polygon contained within a 7.5-minute quadrangle consists of profiles that do not always have the same number of data points because of the variable angle between the UTM coordinate system and true north. The result is a regular, stair-stepped arrangement of the data points along the edges of the map that can translate to blanked grid nodes along the edges of grid files produced in **Surfer**.

Wireframe Overlays

Wireframes can be created independently of other maps, or they can be combined with other maps in using **Map | Overlay Maps**. Wireframes cannot be overlaid with other wireframes or on raster-based maps such as surfaces, image maps, and shaded relief maps. Wireframes can be overlaid with contour maps and other map types.

Chapter 13

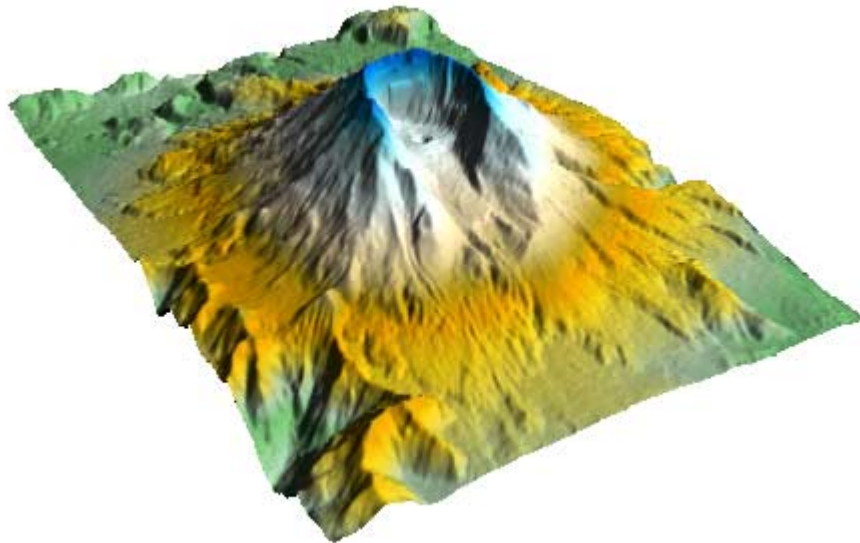
Surfaces

Introduction to Surfaces

A surface is a three-dimensional shaded rendering of a grid file. The height of the surface corresponds to the Z value of the associated grid node. Denser grids show greater detail on the surface.

Color can be used to show Z values on surfaces. The colors are blended to form a smooth gradation. Once the color is selected, the lighting can be adjusted, changing the appearance of the map.

Surfaces can be overlaid with bitmaps, vector files, and other surfaces. You can overlay other map types, with the exception of wireframes, on a surface.



This is a surface created with a USGS SDTS file, available as HELENS2.GRD in the SAMPLES directory.


Surface Components

Surface components include the following:

- Color can be a combination of Z information, lighting, and overlay color.
- Color scale bars can be added to show the relation between color and Z values.
- Light source settings can be changed to any direction on the map.
- Mesh lines can be added to the surface.
- If there are overlays, the color can be set to the surface material color, to the overlay color, or blended between the surface material color and overlay color.
- The base consists of the sides of a block drawn from the edge of the surface vertically downward to the grid's minimum Z value. The sides of the base can be highlighted by changing the color of the fill and lines, from the surface to the bottom of the base.
- Map scale controls the size of the map and the number of map units per page unit.
- Map axes display tick marks and axis labels along the borders of the map. Axes show the X, Y, and Z scaling in map units.
- Map orientation is defined as the rotation, tilt, and field of view of the surface.
- Map limits are the XY extents of the surface.
- A background can be added to a map. The map background is blended with the surface color.

Creating a New Surface

To create a surface:

1. Click **Map | Surface** or click the  button.
2. Select a grid file in the **Open Grid** dialog and click the *Open* button.
3. The surface is created.

Editing Surface Properties

To change the features of the surface, open the surface properties dialog by:

- double-clicking on the map in the plot window,
- double-clicking on the map name in the **Object Manager**,
- right-clicking the map and choosing **Properties** in the context menu,
- or by clicking **Edit | Properties** when the surface is selected.

Surface Properties

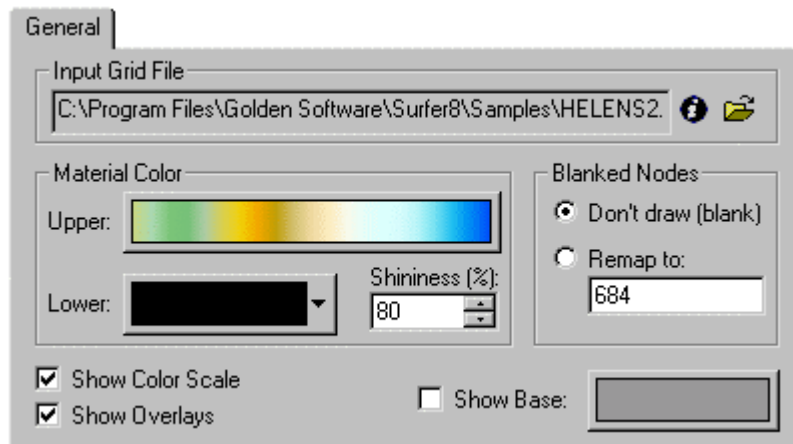
The surface properties dialog contains eight pages: **General**, **Mesh**, **Lighting**, **Overlays**, **Scale**, **View**, **Limits**, and **Background**. **View**, **Scale**, **Limits**, and **Background** are discussed in *Chapter 14, Common Map Properties*.

General

The surface properties **General** page contains the following options:


Grid File

The *Input Grid File* shows the current grid file used in the surface.




*The **General** page controls the surface color, the display of the color scale bar, overlays, blanked areas, and base.*



The  button displays information about the grid file used to produce the surface. The information includes the grid size, the minimum and maximum XYZ values contained in the grid file, and statistics. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.



The  button displays the **Open Grid** dialog. This allows you to open a new grid file or an updated version of the grid used to create the surface. When the grid file is changed, the map limits are reset but the Z data limits for the surface colors are not recalculated. If the Z range for the new grid is outside the old Z range, the full color spectrum may not be used. Click the *Upper* button in the *Material Color* group, and then check the *Use data limits* box to use the full color spectrum. If the new grid ranges are significantly different, it may be better to create a new surface rather than update the grid in an existing surface.

Material Color

The *Material Color* group shows the spectrum of colors used to color the upper and lower sides of the surface. The light angle and light colors also affect the appearance of the surfaces.

- To change the surface material colors, click on the *Upper* button. Several predefined color spectrum files are available in the SAMPLES directory, or you can create your own color file in the **Color Spectrum** dialog. See *Chapter 16, Adding Color to Maps* for more information on the **Color Spectrum** dialog.
- The *Lower* button controls the color of the bottom side of the surface. Unless the bottom of the surface is at a sufficient angle to the light source, there will be insufficient reflection to see the lower color. However, as you increase the ambient light (colors closer to white) the lower surface color becomes more visible because ambient light is everywhere and directionless. The bottom side of the surface uses a single color, not a color spectrum.
- *Shininess (%)* controls the size of the specular reflections. As the shininess percentage increases, the reflections become more focused. See the *Light Color* section for more information about specular reflection.

Blanked Nodes

There are two choices for blanked nodes, *Don't draw (blank)* and *Remap to*. If *Don't draw (blank)* is selected, the blanked areas are not drawn on the map and appear as small transparent gaps in the surface. Alternatively, you can choose to remap the blanked values to a new value. Click the *Remap to* option and then type the new Z value. The new value must be within the range of Z values in the current map.

Color Scale

You can check the *Show Color Scale* option to display a color scale bar adjacent to the map. The color scale bar indicates the colors assigned to the Z levels on the map, and the associated Z level values are displayed as labels on the color scale bar.

Overlays

Click the *Show Overlays* box to show the overlays on the surface. For example, if you overlay a post map on a surface, click the *Show Overlays* box to make the post map visible. Multiple surfaces can be combined into a single composite map, and this allows the overlays to be displayed on an individual basis. The appearance of the overlay is specified on the **Overlays** page.

Base

Click the *Show Base* box to fill the base from the edge of the surface to the axis lines. Click the color box to the right of *Show Base* to change the base color. See the *Surface Base* section for more details.

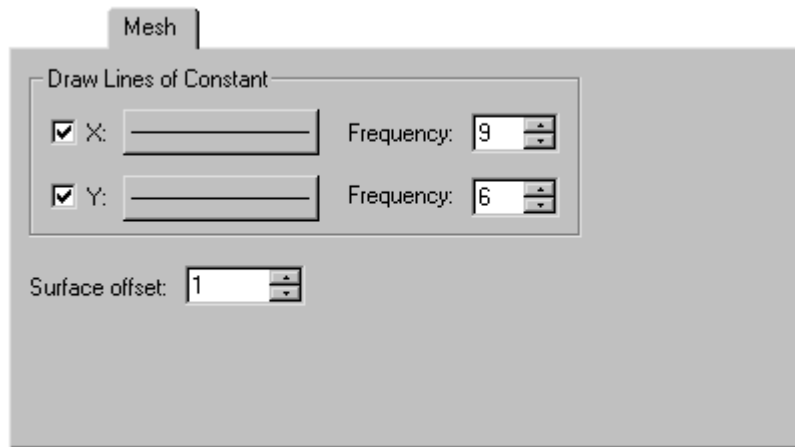
Mesh

You can use the **Mesh** page to draw mesh lines on the surface.

To draw mesh lines:

1. Open the surface properties using one of the methods in the *Editing Surface Properties* section.
2. Check the *X* and *Y* boxes on the **Mesh** page.
3. Click the line buttons to the right of the *X* and *Y* boxes to change the line properties.
4. Change the frequency of the lines by entering a new number into the *Frequency* boxes.

5. If you overlay two or more surfaces, solid mesh lines may appear dashed or broken. Experiment with the *Surface offset* values to change the distance above the surface that the mesh is drawn.



You can check the X and Y boxes to add a mesh to the surface.

Mesh Tips

- The default line frequency is 50 lines in each direction.
- The mesh offset value o is calculated by:
$$o = (m)(\text{factor}) + r$$

where

m maximum depth slope of the polygon

factor user-specified value in *Surface offset*, values typically range from 0.0 to 5.0

r the smallest value guaranteed to produce a resolvable difference in window coordinate depth values (a constant)

Lighting

The **Lighting** page controls the lighting for the entire composite map. This includes the surface and any overlays that may have been combined with the surface. The light source is fixed relative to the surface, so if the surface is rotated, the light rotates with it.

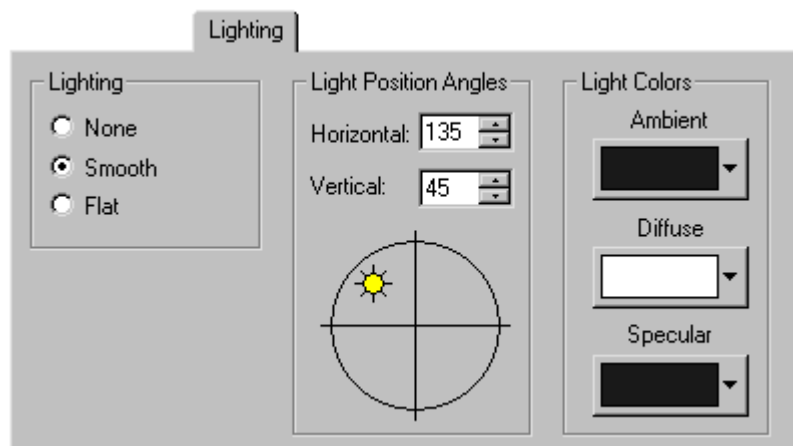
Lighting

There are three lighting options.

- *None* disables all lighting effects. The color shown is from the surface material color only.
- Each grid cell is divided into two triangular polygons. With *Smooth* lighting, Gouraud shading is used to interpolate colors within the triangles from the three vertices of each triangle. This results in smooth transitions across the triangles and the entire grid, but it is slightly slower than flat shading.
- Only one vertex (grid node) is used to define the shaded color for the entire polygon with *Flat* lighting. Note that each grid cell is divided into two triangular polygons. This results in a faceted look since each triangle is only filled with a single color.

Light Position

The *Light Position Angles* group controls the orientation of the light source. The light source can be thought of as the sun shining on a topographic surface. The graphic image shows the relative position of the light source.



You can set the lighting type, angle, and colors on the surface properties **Lighting** page.

Chapter 13 - Surfaces

The *Horizontal* box defines the direction for the light source in the horizontal plane. Zero degrees corresponds to the light source shining from due east. Positive angles rotate the light source counterclockwise. For example, a 90-degree angle places the light source located due north of the unrotated surface. The default *Horizontal* angle is set at 135 degrees, or NW.

The *Vertical* box rotates the light source in the vertical plane. A vertical angle of zero degrees places the light source at the horizon. An angle of 90 degrees places the light source directly overhead. You can also specify negative angles, where the light source originates from under the surface. The default *Vertical* angle is 45 degrees. As the vertical angle approaches zero, shadows lengthen and the colors appear darker.

Light Color

There are three different types of light color, *Ambient*, *Diffuse*, and *Specular*. Note that these colors are used to represent reflectivity. White is 100% reflective and reflects the material color unaltered to the viewer. Black is 0% reflective, and causes all material color to be absorbed. Our perception of color is based on reflected and absorbed light. For example, a leaf appears green because it absorbs all colors in the light spectrum EXCEPT green. Since only green is reflected to your eye, the leaf appears green.

Surfer uses a pure white light source. The light "strikes" the surface and some of the light is absorbed based on the color of the surface material at the point the light ray struck it. Some light is reflected to the viewer according to the type of light (*Ambient*, *Diffuse*, and *Specular*), and the reflectivity color associated with each type of light (specified in the *Light Colors* group).

Ambient refers to light that has been scattered so evenly by the environment that its direction is impossible to determine. Increasing the ambient light component brightens the scene without casting shadows. The default *Ambient* color is 90% black which means that the ambient light contribution is fairly small.

Diffuse refers to light coming from a particular direction and is brighter if aimed directly down on a surface than barely glancing off the surface. When diffuse light hits the surface, it is scattered uniformly in all directions so that it appears equally bright no matter where the eye is located. Increasing diffuse light intensifies shadow effects. The default *Diffuse* color is white, which is the maximum amount of reflectivity.

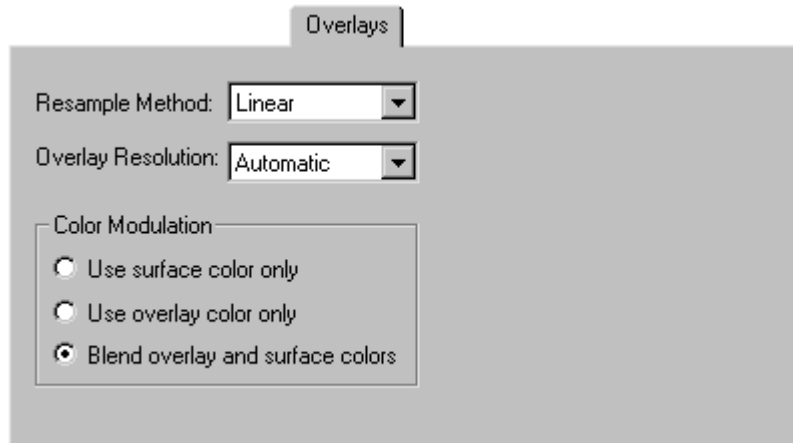
Specular refers to light coming from a particular direction, and tends to bounce off the surface in a preferred direction. A shiny surface such as metal has a high specular component, while a surface like carpet has almost no specular component. Increasing the percentage of specular light results

in strong shadow effects and more pronounced "shiny" or glare spots. The default *Specular* color is 90% black.

In general, these reflectivity colors should be specified as shades of gray in order to evenly reflect the surface material color components. However, special effects are possible by specifying non-gray colors for the reflectivity. For example, assume the *Ambient* reflectivity is set to pure red, and the *Diffuse* and *Specular* components are set to pure black. The *Diffuse* and *Specular* components are essentially disabled by setting their reflectivity color to black. The only light that is reflected to the viewer is red ambient light. Portions of the surface that lack a red component in the material color will appear black, since only red light is reflected to the viewer.

Overlays

The **Overlays** page contains options to control how overlays are combined with the surface. Overlays are converted into a bitmap known as a texture map. This texture map is applied to the surface by stretching it and shrinking it as necessary. Note that this stretching applies to all graphics in the overlays, including text. If the text is positioned over a steep portion of the surface, it can be stretched quite a bit, resulting in significant distortion. The usual solution to this is to view the surface from a higher angle to minimize stretching along the Z axis.



*You can show only the surface color, only the overlay color, or you can blend the surface and overlay colors using the **Overlays** page.*

Resample Method

When the texture map is stretched, the colors in the original overlay must be resampled to a new size and position. The *Resample Method* specifies how the texture map is resampled.

- *Linear* uses bilinear interpolation to combine the four surrounding pixels. Bilinear interpolation results in higher quality, but is usually slower.
- *Nearest* uses the nearest pixel in the source image.

Overlay Resolution

The *Overlay Resolution* refers to the resolution of the texture map. High-resolution texture maps result in more detail, but line and text features become thinner and may eventually fade or break up.

Color Modulation

Color Modulation refers to the method used to combine the texture map and surface material colors. You can use only the surface material colors (**General** page), use only the overlay colors (such as contour map colors or bitmap colors), or you can blend the surface and overlay colors. To display colors from an overlay with surface shadows but without surface colors, set the surface color to a uniform white color using the **General** page *Upper Material Color*.

Overlay Visibility

If there are overlays on a surface, and the surface is made invisible through the **Object Manager**, the overlays will not be visible. Instead of making the surface invisible, you can change the *Color Modulation* to *Use overlay color only*.

Surface Base

The surface base is the vertical area from the edge of the surface extending downward to the plane of the XY axes at the grid's minimum Z value. The base fill, line, and lighting properties can be set. When two surfaces of the same XY limits are overlaid, and the bases have different solid colors, the two colors "stitch" together.

To set the surface base properties:

1. Open the surface properties using one of the methods listed in the *Editing Surface Properties* section.
2. Check the *Show Base* box on the **General** page.
3. Click the color button to the right of the *Show Base* box.
4. Set the fill and line properties in the **Fill and Line Properties** dialog.
5. Click OK in the **Fill and Line Properties** dialog to return to the surface properties dialog.
6. Click OK or *Apply* in the surface properties dialog to apply the base properties to the map.



*On the **General** page, click the button to the right of Show Base to set the base properties.*

Base Properties

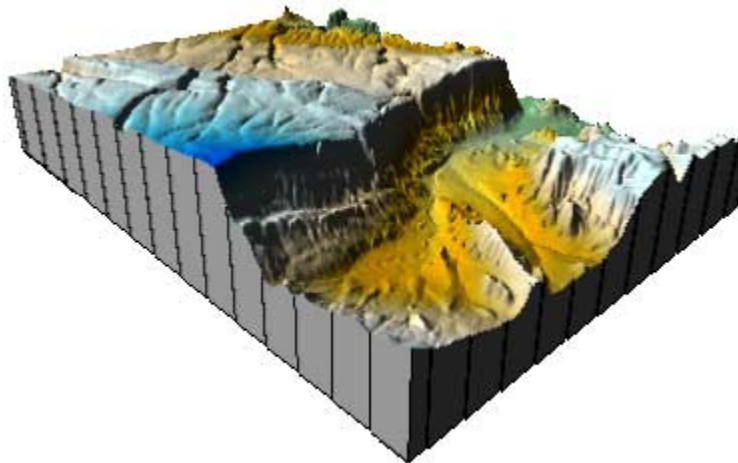
The *Fill* group contains options for filling the surface base.

- Choose *None* from the *Style* list if you do not want to fill the base.
- Choose *Solid* from the *Style* list to fill the base with a solid color.
- Choose *Lighted* from the *Style* list to fill a base with a solid color and apply the lighting properties to the sides of a base.
- Click the *Color* button to set the color of the base fill.

Chapter 13 - Surfaces

The *Line* group controls the outline of the base fill.

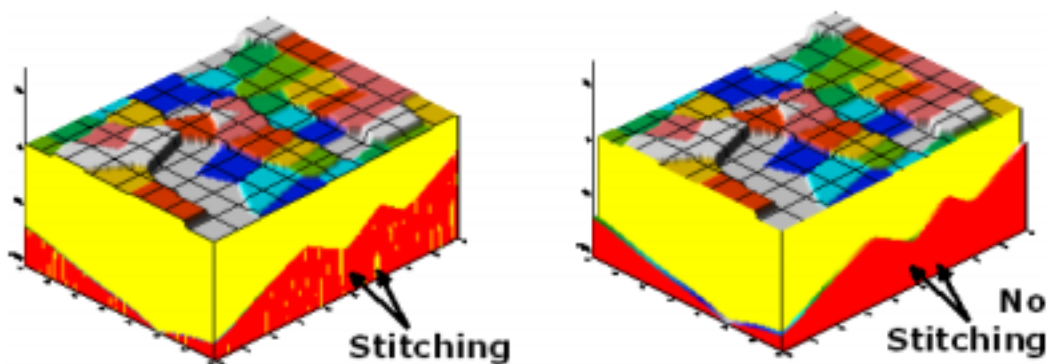
- Choose *None* from the *Style* list if you do not want to outline the base.
- Choose *Solid* from the *Style* list to outline the base with a solid color.
- Click the *Color* button to set the color of the base outline.
- Enter a *Width* value from 0.0 to 0.5 inches (1.27cm) for the base outline.



This is a surface with Show Base enabled. The fill color is lighted, and since the light is coming from the upper left part of the map, the lower right base fill is a darker color.

Surface Overlays, Bases, and "Stitching"

When multiple surfaces with the same X and Y minimum and maximum are overlaid, and the surfaces' bases have a solid fill, the lower base may have a mix of two or more colors due to "stitching" where the bases coincide. This effect changes based on the zoom percentage and resolution of the output device, and might not be visible when printed or exported to a file. If stitching does occur, you can add an offset to display unbroken fills. To change the base appearance, you can grid the upper surface with slightly larger minimums and slightly smaller maximums, or use the **Grid | Extract** menu to remove the first and last rows and columns of the grid.



Stitching occurs when multiple filled bases are coincident. For unbroken base fills, you can add an offset to the upper grid by regriding or extracting a subset of the grid.

Surface Memory Requirements

Surfaces and Zoom Levels

Surfaces use a large amount of memory to zoom in to a small area on the map. For example, a 5 MB surface map can require 150 MB of memory when zoomed in. You can reduce the memory requirement by reducing the number of *Undo Levels* through the **File | Preferences | General** command.

Surfaces and Printing

Surfaces may require a large amount of memory when sending data to a print spooler. For example, a 187 KB SRF file containing a surface requires 29.9 MB of memory in the print spooler. To reduce the amount of memory required, select the *Print directly to printer* option in the printer properties. This option reduces the spooler memory requirement in the above example to 739 KB.

Color Scale Bars

A color scale bar is a legend that shows the colors used in the surface.

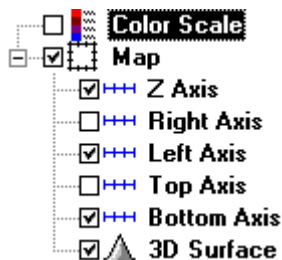
Adding a Color Scale Bar

To add a color scale bar to a surface:

1. Open the map properties using one of the methods listed in the *Editing Surface Properties* section.
2. Specify the colors to use in the map on the **General** page.
3. Check the *Show Color Scale* box to activate the color scale bar.
4. Click OK or *Apply* to draw the map with a color scale bar.

Color Scale Bar Visibility

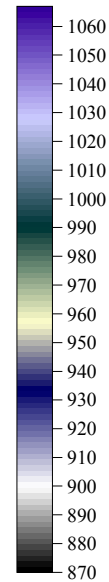
The color scale bar can be toggled on and off through the **Object Manager**.



*Remove the check mark from the Color Scale box in the **Object Manager** to make the color scale bar invisible.*

Removing a Color Scale Bar

A color scale bar can be removed by clicking on the scale bar and pressing the DELETE key on your keyboard. Alternatively, you can uncheck the *Show Color Scale* check box in the map properties dialog.

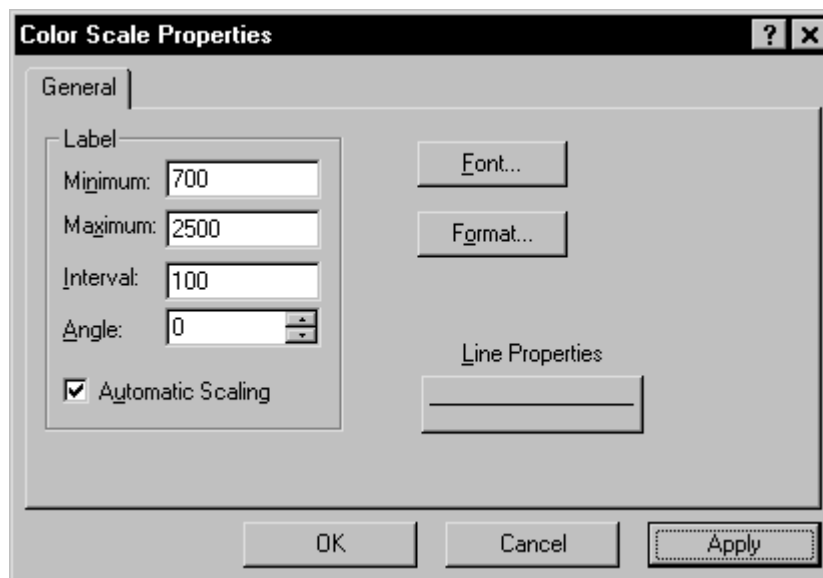


Color scale bars are added through the surface properties dialog.

Editing a Color Scale Bar

To modify the color scale bar:

1. Double-click the color scale bar to display the **Color Scale Properties** dialog.
 - Set the first label value in the *Minimum* box.
 - Set the last label value in the *Maximum* box.
 - Define the spacing between labels in the *Interval* box.
 - Choose the angle at which the labels are drawn in the *Angle* box. Positive angles rotate labels in a counterclockwise direction.
 - The *Automatic Scaling* check box sets the label frequency so labels do not overlap. If there are too many labels to fit into the space next to the color scale bar, some labels are dropped to make the labels legible.
 - Click the *Font* button to set the label text properties.
 - Click the *Format* button to choose the label numeric format (i.e. number of decimal places).
 - The *Line Properties* button allows you to specify the line properties for the color scale bar border.
2. Once the changes are made, click OK or *Apply* to redraw the color scale bar.



The *Color Scale Properties* dialog controls the display of the color scale bar.

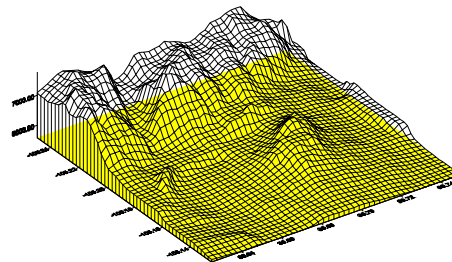
Chapter 14

Common Map Properties

Introduction to Common Map Properties

Properties common to all overlays in a map are located in the map properties dialog. These properties include the view, scale, limits, and background. If there are multiple overlays in a map, the changes made on the **View**, **Scale**, **Limits**, and **Background** pages apply to all overlays.

- Maps can be displayed at any orientation using the **View** page. The tilt, rotation, field of view, and projection of a map are changed via this command.
- Map scale controls the size of a map on the printed page. This is accomplished by defining a correspondence between length on the map (in map units) and length on the printed page (in page units). For example, one mile on the map corresponds to one inch on the page. The map scale is specified along the map axes using the **Scale** page. The map scale for the X axis and the map scale for the Y axis can be set in unison (proportionally), or they can be set independently. On wireframes and surfaces, the map scale for the Z axis can also be set.
- Map limits are defined as the X and Y extent of the map. When a map is first created, the map limits are set to the minimum and maximum X and Y coordinate values in the grid file, base map, or data file used to create the map. You can accept these as the limits for the map, or you can specify custom limits using the **Limits** page. Custom limits are useful when plotting only a portion of a map.
- In addition to adding color to maps as discussed in other chapters (i.e. contour fill), a background color can be added to the wireframe. The **Background** page sets the background fill properties and the border line properties for a map. The map background limits coincide with the axis limits on contour, base, post, and vector maps. On wireframes, the map background refers to the bottom plane of the base. On surfaces, the background color is blended with the surface.



The map background on a wireframe refers to the bottom plane of the base.

Opening Map Properties

The properties of one or more maps can be opened by one of the following methods:

- Select the **Edit | Properties** command when the maps are selected
- Double-click on the object in the plot window or in the **Object Manager**
- Right-click on an object in the plot window or in the **Object Manager** and select the **Properties** command

View

Maps can be displayed at any orientation using the map properties **View** page or using **Map | Trackball**. The tilt, rotation, field of view, and projection of a map can be changed.

View and Overlays

When you combine maps using the **Map | Overlay Maps** command, the resulting map uses the view parameters from the wireframe or surface.

When you have more than one map on the page, the view settings are applied to all selected maps. For example, this command is useful if you have a contour map and wireframe based on the same grid file and displayed in the same plot window. You can select the contour map and wireframe, open the properties with **Edit | Properties**, and then choose the **View** page. Any values specified are applied to both maps.

Field of View

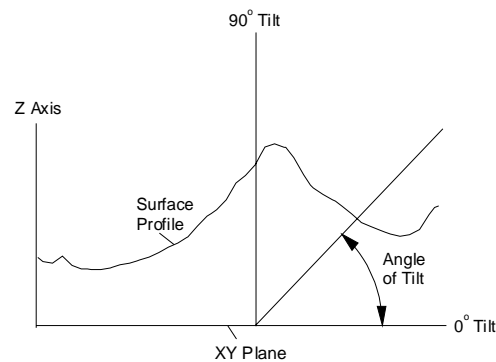
The field of view controls the perspective effect of the perspective projection option. The field of view angle is formed by the left edge of the bounding box, the viewpoint, and the right edge of the bounding box. As the field of view angle is increased, the viewpoint moves closer to the map, and the perspective effect becomes more pronounced. With an orthographic projection, the field of view values have no effect. Note that the field of view parameter does not influence the map scale.

Rotation

Rotation controls the rotation of the map about the Z axis. As the rotation value increases, the map rotates in a counterclockwise direction.

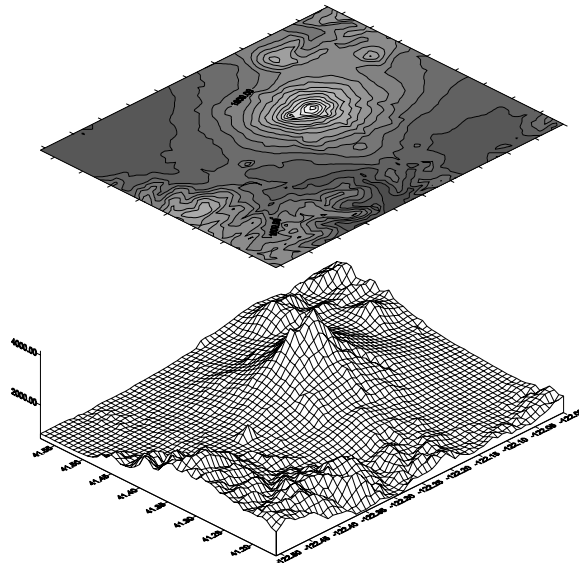
Tilt

Tilt is defined as the angle between the viewing direction and the XY plane of the map. A 90 degree tilt means the XY plane lies in the plane of the page or screen and is orthogonal to the viewing direction. As the angle of tilt increases, the Z axis tilts toward the viewer. When you tilt three dimensional maps (e.g. wireframes) to 90 degrees, the surface relief is no longer visible. As the angle of tilt decreases, the XY plane of the map is viewed more on-edge. The default tilt is 30 degrees for wireframes and surfaces, and 90 degrees for all other map types.



Projection

The projection sets the orthographic or perspective projection for the selected maps. The perspective option creates a visual effect whereby the size of the surface varies with the distance from the observer. When a perspective projection is used, you can change the field of view value to increase or decrease the perspective effect. The orthographic option creates a plot with a projection of the surface onto a plane, oriented perpendicular to the line of sight between the observer and the surface. Perspective projections more closely simulate reality, but orthographic projections maintain parallel lines and allow accurate measurements to be taken from the drawing.

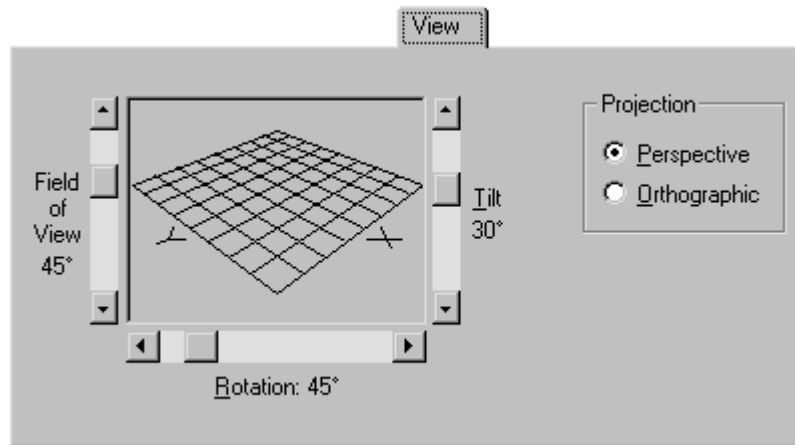


Maps can be displayed at any rotation angle and at any tilt angle from zero to 90 degrees. When more than one map is selected, the view parameters are applied to all selected maps.

Setting the View

To modify the map orientation with the **View** page:

1. Select the map or maps that you want to view at a different orientation.
2. Open the map properties using one of the methods listed in the *Opening Map Properties* section of this chapter.
3. Click the **View** page.
4. The graphic on the **View** page changes to show the orientation as you adjust the *Field of View*, *Rotation*, *Tilt*, and *Projection* parameters.




The View page controls the orientation of the map or maps.

5. Click OK or Apply and the selected maps are drawn at the new orientation.

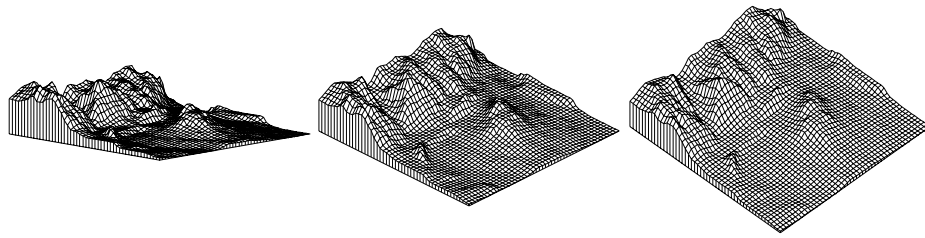
Using the Trackball Command

The **Map | Trackball** command allows you to rotate, tilt, and change the field of view of a selected map on screen.

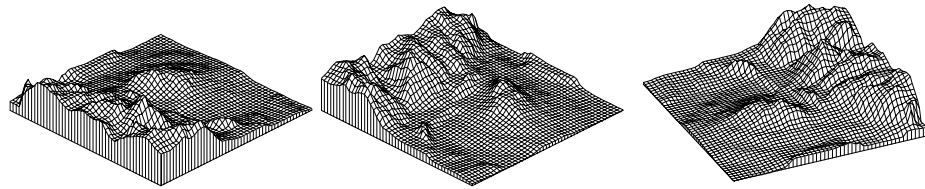
To modify the map orientation with **Trackball**:

1. Select one map.
2. Select **Map | Trackball** or click the  button.
3. Left click inside the map's bounding box. While holding down the left mouse button, drag the mouse up and down to change the tilt, and drag the mouse left or right to change the rotation. The rotation and tilt values are shown in the status bar.
4. If you right click in the map's bounding box, dragging up increases the field of view and changes the *Projection* to *Perspective*. Dragging down while right clicking decreases the field of view, and the *Projection* changes to *Orthographic* if the field of view is decreased to 1°. The field of view value is shown in status bar.

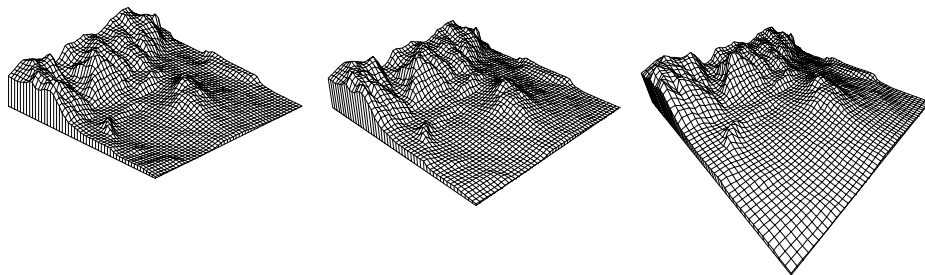
Chapter 14 - Common Map Properties



This is the same wireframe displayed at the same rotation angle but different tilt angles: 10, 30, and 45 degrees.



This is the same wireframe displayed at the same tilt angle but different rotation angles.



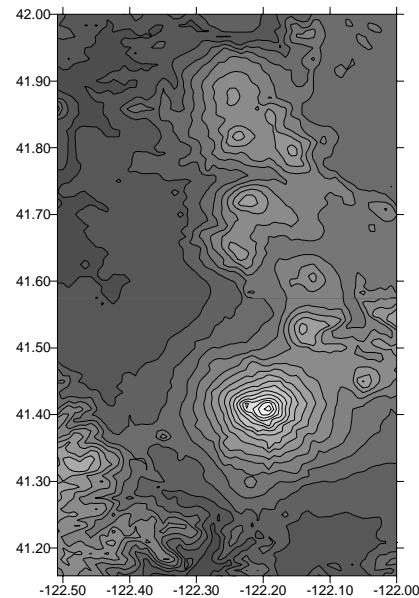
This is the same wireframe using different projection parameters. The map on the left uses an Orthographic projection. The wireframe in the middle uses a Perspective projection with a Field of View value of 20°. The plot on the right uses a Perspective projection with a Field of View value of 75°.

Scale

By default, map scales are defined proportionally in the X and Y dimensions, with the longest side measuring six inches on the printed page. For example, consider a map extending 40 map units in the X dimension and 60 map units in the Y dimension. Since the extent in the Y dimension is longer than the extent in the X dimension, the map is drawn six inches high. The resulting map scale is 10 map units per inch. The map scale in the X dimension is also 10 map units per inch, so the map is drawn four inches wide.

On three-dimensional maps, such as wireframes, the default map scale in the Z dimension is defined such that the vertical extent of the map is one-fourth the maximum horizontal extent.

The default map scale is a function of map limits. Map limits are determined by the file used to create the map, or they can be defined using the **Limits** page. For more information on redefining map limits, see the *Limits* section of this chapter.



The Y dimension covers a greater extent than the X dimension in this map. By default, the Y axis would be six inches in length, and the X axis is proportionally shorter.

Setting the Scale

A map can be scaled in the X, Y, and Z directions using the **Scale** page in the map properties dialog.

1. Select a map.
2. Open the map properties using one of the methods listed in the *Opening Map Properties* section of this chapter.
3. Click the **Scale** page.
4. Set the scale using the following fields:
 - The *X Scale*, *Y Scale*, and *Z Scale* groups set the map scale along the specified axes. Each axis can be scaled differently.
 - The *1.0 in = ___ Map units* box specifies the number of map units per page unit in the associated dimension. For example, to draw a map at a scale of 1000 map units per inch, type the value 1000 into the box. The *Length* value is automatically updated to reflect the change. For example, if the map is 8000 units in the dimension you are setting, the map is eight inches long in that dimension.

Chapter 14 - Common Map Properties

- The *Length* box specifies the length of the map in the X, Y, or Z dimension. When the *Length* value is changed, the *1.0 in. = ____ Map units* box is automatically updated to reflect the change.
- The *Proportional XY Scaling* check box ensures that the X and Y dimensions are scaled equally. When the check box is activated, a change in either the X or Y dimension is reflected in the other dimension by a proportional amount. To scale the axes independently, turn off the check box.

The screenshot shows a dialog box titled "Scale". It contains three sections for X Scale, Y Scale, and Z Scale. Each section has a "1.0 in. =" input field, a "Map units" label, and a "Length:" input field with a spinner. The X Scale section shows "2360" for 1.0 in. and "4.78 in" for Length. The Y Scale section shows "2360" for 1.0 in. and "6.00 in" for Length. The Z Scale section shows "645.3333333" for 1.0 in. and "1.50 in" for Length. At the bottom, there is a checked checkbox labeled "Proportional XY Scaling".

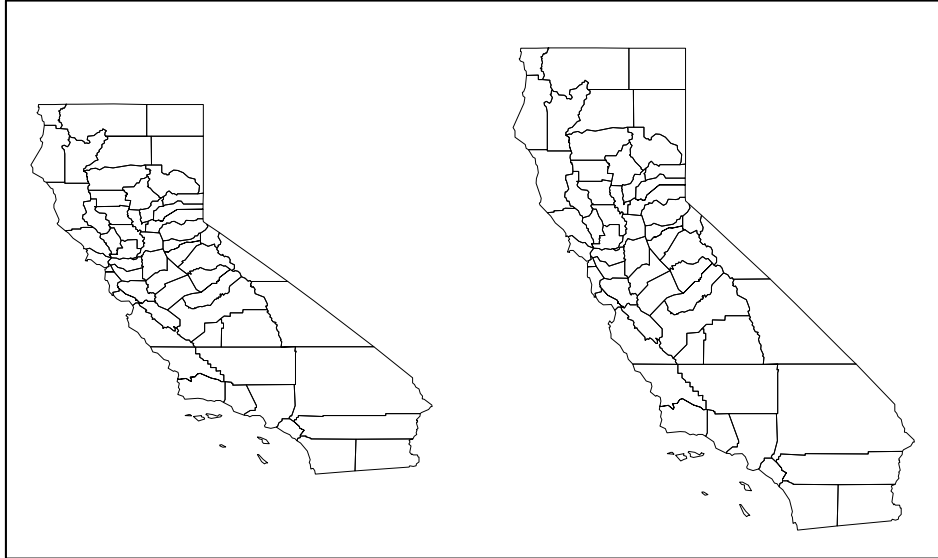
The Scale page allows you to specify the relative X, Y, and Z scaling.

5. Change any of the scale parameters by entering the new values into the boxes.
6. Click the OK button to draw the map at the specified scaling.

NOTE: Sizing a map with its selection handles does not rescale the map internally; it just resizes the map on the screen. To return to the proper scale, use the **Arrange | Transform** command to undo the sizing, and then use the **Scale** page to scale the map.

Using Scaling to Minimize Distortion on Latitude/Longitude Maps

When you display maps based on latitude/longitude coordinates extending over a large region, they might appear somewhat distorted. This occurs because one degree of latitude is not equivalent to one degree of longitude in distance. For example, consider a base map of the state of California (CA.GSB).



This shows the map of California before and after scaling. The map on the right is scaled up in the Y dimension so the map does not appear compressed.

To determine the scaling factors:

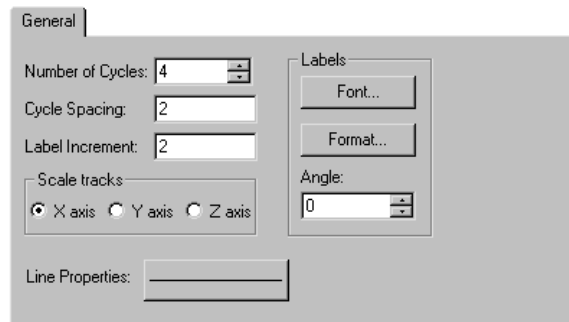
1. Determine the latitude for the parallel through the center of the map, and obtain the cosine for this latitude value. The center parallel in the sample file CA.GSB is approximately 37.27 degrees. The cosine of 37.27 is 0.80.
2. Open the map properties using one of the methods listed in the *Opening Map Properties* section of this chapter.
3. Click the **Scale** page and set the X direction for the map. You can set either the *Length* value or the *Map units* value. The *X Scale Map units* is 1.72 in this example.
4. Uncheck the *Proportional XY Scaling* check box.
5. Multiply the *X Scale Map units* value by the cosine of the latitude, and enter this number into the *Y Scale Map units* field. $1.72 (X \text{ Scale Map units}) \times 0.80 (\text{cosine of latitude}) = 1.38 (Y \text{ Scale Map units})$
6. Click OK or Apply and the map is drawn with the new scaling.

Adding a Scale Bar to a Map

The **Map | Scale Bar** command places a scale bar on the map. Scale bars are divided into equally spaced cycles. Any number of cycles between one and 100 can be used. Each cycle is labeled based on map units. You have flexibility over the spacing of the cycles and the number of map units the cycles represent on the scale bar. If the defaults are accepted, the scale bar is drawn relative to X dimension map units.

To place a scale bar on a map:

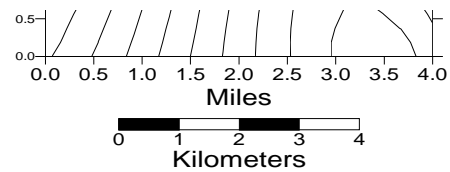
1. Select the map.
2. Choose the **Map | Scale Bar** command to open the **Scale Bar** dialog.
3. Specify the dimension on which you want to base the scale bar by selecting the appropriate button in the *Scale tracks* group. If the X and Y dimensions are scaled proportionally, the *X axis* and *Y axis* options have the same result. The *Z axis* option is only appropriate for wireframes, surfaces, and tilted post maps with *3D Label Lines*.
4. Specify the *Number of Cycles* to use for the scale bar. This is the number of divisions the scale bar uses, and can be any value from one to 100.
5. Specify the *Cycle Spacing* by entering the value into the box. Cycle spacing determines the spacing, in map units, for each cycle on the scale bar. For example, if you specify a *Cycle Spacing* of 5, then each cycle on the scale bar is five map units long, and the scale bar is a total of 20 units long if the *Number of Cycles* is set to 4.
6. The *Label Increment* specifies the change in label value from the beginning of one cycle to the beginning of the next. Usually the *Label Increment* value is equal to the *Cycle Spacing* value, although you can specify label increments different from the *Cycle Spacing*.
7. Label properties are controlled from the *Labels* group. The *Font* button opens the **Text Properties** dialog so you can change the font, points, style, and color of the text. Click the *Label Format* button to display the **Label Format** dialog, in which you specify the numeric format used for the labels. You can rotate the labels by entering a number into the *Angle* box.
8. You can specify the line properties for the scale bar by clicking on the *Line Properties* button.
9. Click OK and a scale bar is drawn for the selected map.



The scale bar properties are set in the **Scale Bar** dialog.

Scale Bar Example

Consider a map with an axis scaled in miles, but you want to show a scale bar using kilometers. To accomplish this, set the *Cycle Spacing* to 0.62137 (the number of miles per kilometer). This cycle spacing value results in a cycle that is one kilometer long relative to the map. Then, set the *Label Increment* value to 1, representing one kilometer per cycle.



In this example, the map axis is scaled in miles, and the scale bar is scaled in kilometers.

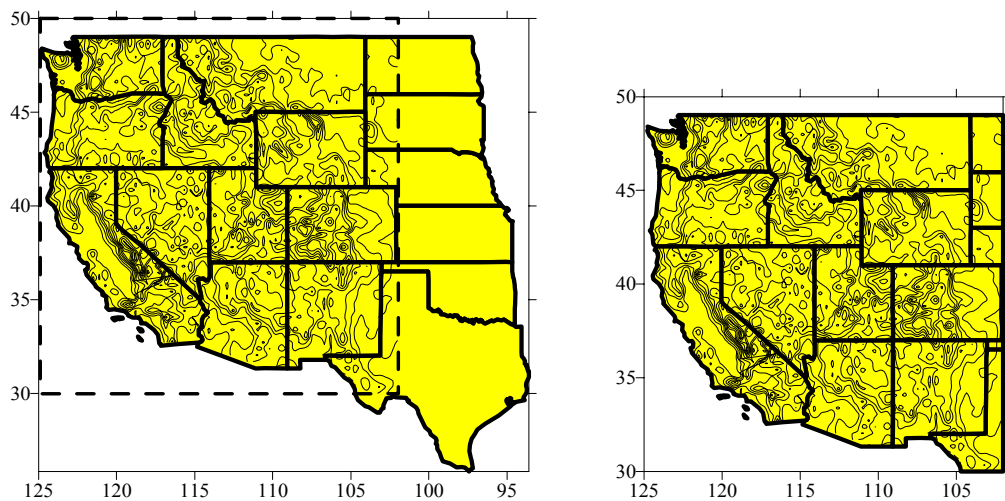
The screenshot shows the 'Scale Bar' dialog box with the 'General' tab selected. The 'Number of Cycles' is set to 4. The 'Cycle Spacing' is set to 0.62137. The 'Label Increment' is set to 1. The 'Scale tracks' section has three radio buttons: 'X axis' (selected), 'Y axis', and 'Z axis'. The 'Labels' section has two buttons: 'Font...' and 'Format...'. The 'Angle' is set to 0. The 'Line Properties' section has a button labeled 'Line Properties:'. The 'Scale tracks' section also includes a 'Scale tracks' label and three radio buttons: 'X axis' (selected), 'Y axis', and 'Z axis'.

*The **Scale Bar** dialog shows the Cycle Spacing and Label Increment values used to generate the scale bar shown above.*

Limits

When a map is created, the limits are automatically defined by the coordinate values contained in the file used to create the map. The **Limits** page in the map properties dialog lets you redefine the X and Y minimum and maximum map coordinates (specified as *xMin*, *xMax*, *yMin*, and *yMax* values). The **Limits** page is unavailable if there is a wireframe in the map.

Limits can be set larger or smaller than the limits of the current map. For example, if you have a map that ranges from 0 to 100 in the X dimension, but you only want to display the map from 25 to 50 in the X dimension, use 25 and 50 as the new minimum and maximum values on the **Limits** page.



In the map on the left, the limits are set to the default limits as defined by the base map, while the limits of the contour map are indicated by the dashed rectangle. The map on the right has been clipped to the limits of the contour map.

Limits and Wireframes

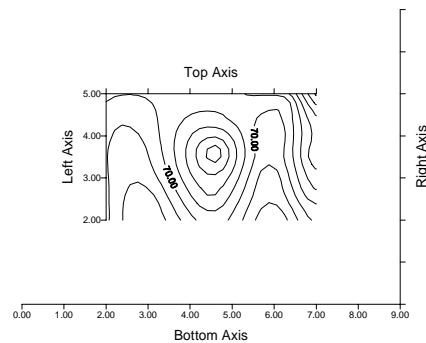
Limits cannot be defined for wireframes and this constraint includes maps containing wireframe overlays. These maps are limited to the extent of the grid file used to create the wireframe. To change the limits of a grid file for a wireframe, use **Grid | Extract** or create a new grid file with new limits.

Limits and Axis Scaling

When you change limits, axes that use automatic scaling are redefined to fit the new limits of the map. If you do not use automatic scaling for the axes, the axes are drawn over the same range as the original map and will not match the map using the new limits.

Limits and Map Scale

When new limits are defined for a map, the map scale is not changed. For example, if the map uses a scaling of 1 inch = 1000 meters, the new map uses this same scale. This can result in maps much smaller or larger than the original map. In this case, open the map properties and set the new scale on the **Scale** page.



The map limits have been reset on this map. The Left Axis and Top Axis use automatic scaling and match the new map limits. The Right Axis and Bottom Axis do not use automatic scaling, and reflect the original limits of the map.

Limits and Post Maps

Limits can be set on post maps or maps containing a post map overlay. When a posted point lies outside the map limits, the posted symbol and the entire label are clipped from the map. When a posted point is at or inside the map limits, the entire symbol and posted label are printed, even if they extend beyond the limits of the map.

Limits and Base Maps

When map limits are set for base maps, or maps containing a base map overlay, the base map boundary lines are drawn up to the limits of the map. When base maps are overlaid on contour maps, it is often best to clip the base map to the limits of the contour map. To do this, select the base map and set the map limits to match the limits for the contour map, thereby clipping the boundaries at the contour map limits.

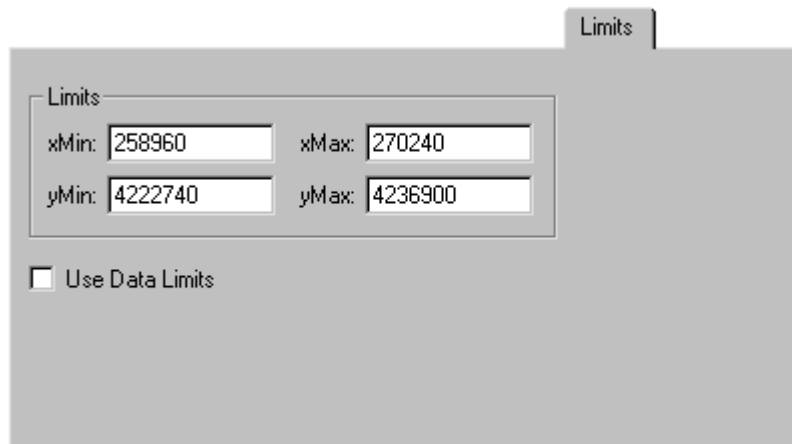
Limits, Scale, and Overlaying Maps

When a new map is overlaid on an existing map, the limits are changed to reflect the minimum and maximum X and Y of all overlays in the map, and the scale is reset to six inches for the longest side. You can set the default scale and limits using **File | Preferences | Dialog Defaults**.

Setting Map Limits

To set map limits:

1. Select a map.
2. Open the map properties using one of the methods listed in the *Opening Map Properties* section of this chapter.
3. Click the **Limits** page.
4. Specify the X and Y minimum and maximum values for the map by entering the values into the *xMin*, *xMax*, *yMin*, and *yMax* boxes. The *Use Data Limits* check box resets the map limits to the limits of the grid file, base map, or posting data file limits.
5. Click OK or *Apply* and the map is redrawn with the new limits. If the axes use automatic scaling, they are also redrawn to fit the new map limits.



The screenshot shows a dialog box titled "Limits" with a tab labeled "Limits". Inside the dialog, there is a section titled "Limits" containing four text input fields: "xMin:" with the value "258960", "xMax:" with the value "270240", "yMin:" with the value "4222740", and "yMax:" with the value "4236900". Below these fields is a checkbox labeled "Use Data Limits" which is currently unchecked.

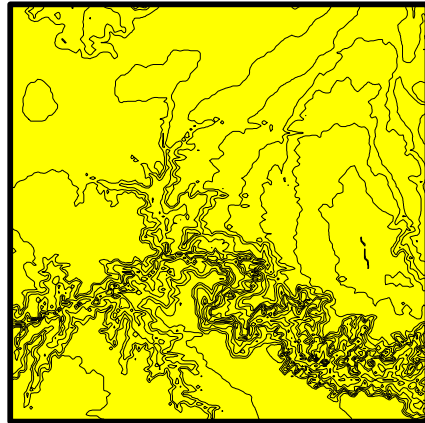
You can set the limits to the file limits by checking the Use Data Limits box.

Background

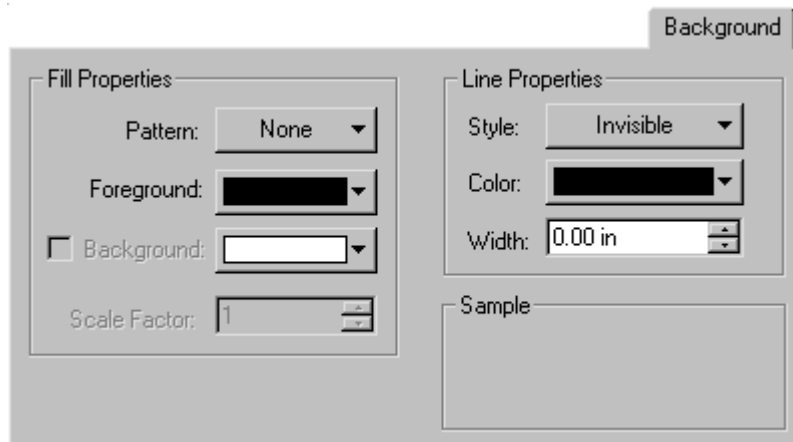
The **Background** page sets the background fill properties and the border line properties for a map. Map backgrounds lie underneath all other objects on the map. This means that the background can be obscured by other map features. For example, when you create color filled contour maps, the fill between contours obscures the map background wherever a solid or bitmap pattern is used for the contour fill. Similarly, the outline of the map background can be obscured by the map axes.

To specify background properties for a map:

1. Open the map properties using one of the methods listed in the *Opening Map Properties* section of this chapter.
2. Click the **Background** page.
3. Set the fill properties by clicking on the *Pattern*, *Foreground*, and *Background* buttons.
4. Click the line properties *Style*, *Color*, and *Width* buttons.
5. Click OK or *Apply* and the properties are applied to the background of the map.



This is a contour map displaying a map background. The axes are not displayed, and a heavy border is added as part of the background.



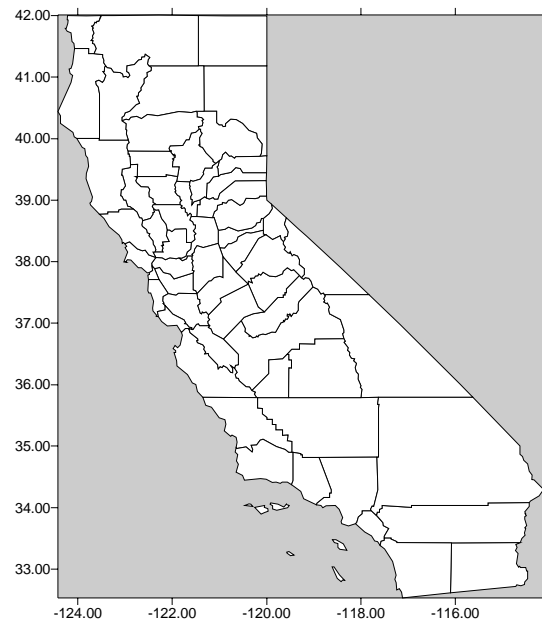
*Select the background fill patterns and colors as well as the line patterns and colors from the **Background** page.*

Masking with Background

Polygons can also obscure the background on a base map. For example, consider the map of California shown here. The fill properties for the counties (polygons) on the base map are set to solid white, and the map background uses a gray background.

To create a map like this one:

1. Create a base map containing polygons.
2. Open the map properties using one of the methods listed in the *Opening Map Properties* section of this chapter.
3. Click the **Base Map** page.
4. Click the *Fill* button.
5. In the **Fill Properties** dialog, choose *Solid* for the *Fill Pattern*, select *White* for the *Foreground* color, and then click OK.
6. Click the **Background** page.
7. In the *Fill Properties* group, choose *50.0% Black* for the *Pattern*, and choose *Black* for the *Foreground* color. Choose *White* for the *Background* color.
8. Click OK or *Apply* to redraw the map with a gray background.



This base map of California uses background color to create the shaded background.

Chapter 15

Axes

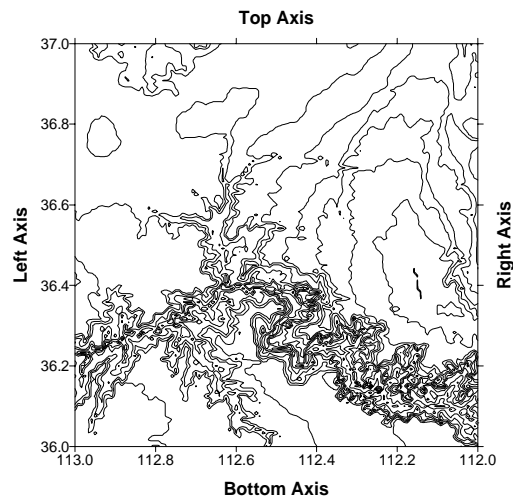
Introduction to Axes

Four axes are automatically created for each map in **Surfer**. Axes are named by their relative positions on the map and are called the Left Axis, Right Axis, Top Axis, and Bottom Axis. The Left Axis and Right Axis are Y axes, and the Top Axis and Bottom Axis are X axes. Wireframes and surfaces also have a single Z Axis.

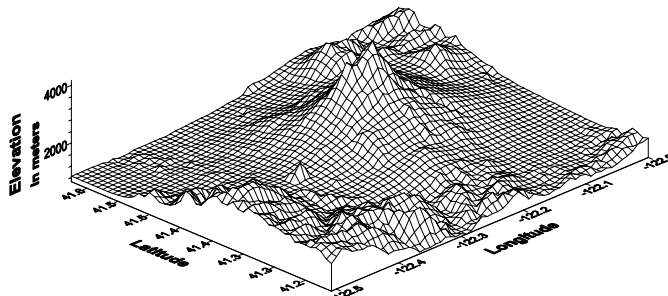
Variograms have two axes with the default names of Y Axis and X Axis.

Axis limits match the XY limits of the file used to create the map. When a map is first created, the Left Axis and Bottom Axis display major ticks and tick labels, and the Top Axis and Right Axis display only major ticks.

The tick spacing is automatically scaled so a reasonable number of ticks are drawn along each axis. Tick scaling and label format parameters can be independently specified for the axes on the map.



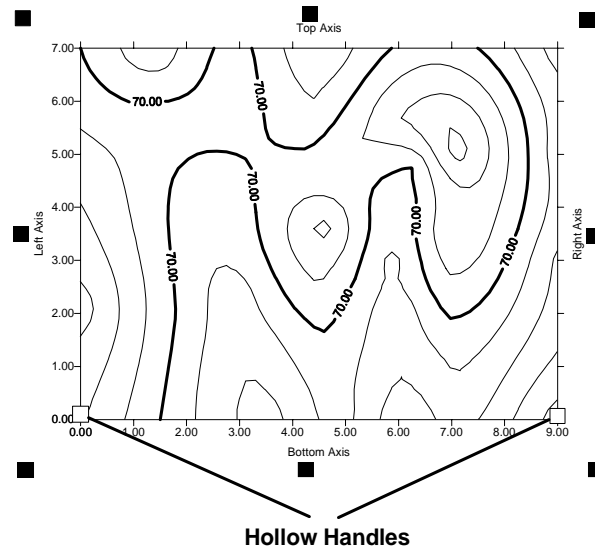
This is a contour map with the four axes labeled with their default names.



A wireframe contains a Z axis in addition to the X and Y axes.

Selecting an Axis

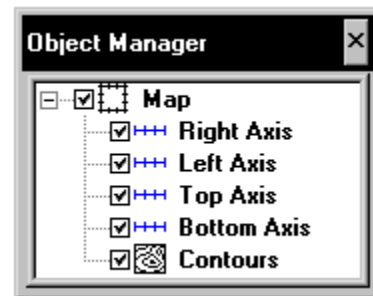
You can click on the axis name in the **Object Manager**, or you can place the cursor on the tick marks or tick labels when clicking. When an axis is selected, hollow square handles appear at each end of the axis. Because the axes are part of the map on which they are drawn, the eight solid handles appear around the entire map. In addition, the name of the axis is displayed on the left side of the status bar. For example, if you select the bottom axis of a contour map, the default text in the status bar appears as "Map: Bottom Axis." The map and axis are also highlighted in the **Object Manager**.



When an axis is selected, hollow handles appear at the ends of the axis (shown on the bottom axis on this map) and solid handles appear around the entire map.

Displaying Axes

When the map or variogram is created, the axes are displayed automatically. You can turn off the display of any or all of the axes by checking or unchecking the axis visibility box in the **Object Manager**. On a wireframe or surface, only the Left Axis, Bottom Axis, and Z Axis are initially displayed.

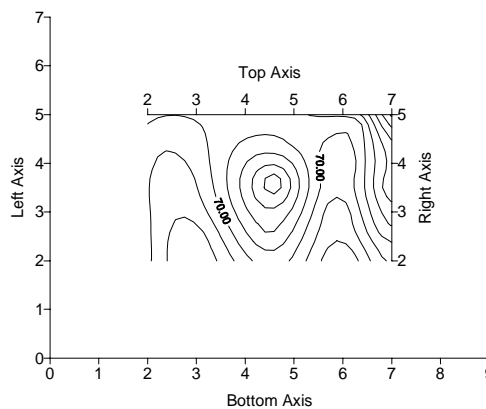


*The axes checked in the **Object Manager** are displayed on the map.*

Axis Scaling

Axis scaling controls the axis limits, the major tick starting and ending values, and the position of the axis relative to the other axes. To set axis scaling, double-click the axis and click the **Scaling** page in the axis properties dialog.

Axis scaling is different from setting the map scale. When axis scaling is set, the axis limits are defined but this does not rescale the map. Axes that use different limits than the map are still drawn at the same scale as the map. Because of this, it is possible to have axes shorter or longer than the map. To change the map scale, open the map properties and choose the **Scale** page. See *Chapter 14, Common Map Properties* for more information on setting the map scale.



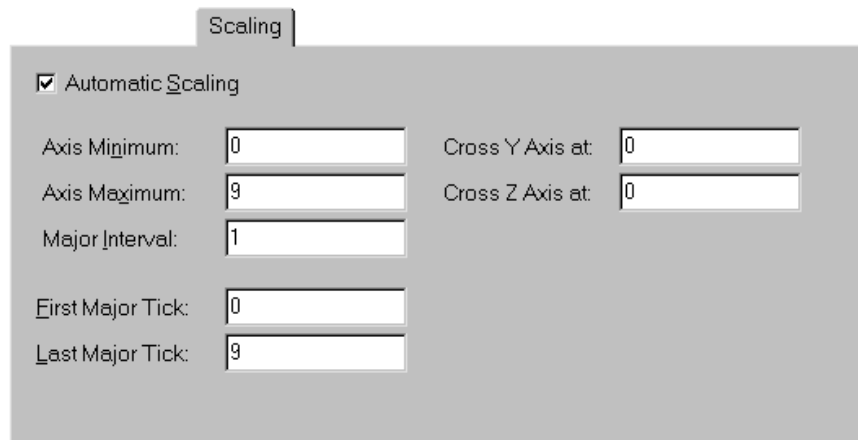
In this example, the Bottom Axis and Left Axis are offset from the map, and the limits of the axes are set differently than the map limits.

Axis Limits

Axis limits are specified in map coordinates or variogram units. Axis limits do not need to coincide with the file limits.

To set axis limits:

1. Double-click on the axis.
2. Click the **Scaling** page in the axis properties dialog.
3. You can set any of the parameters listed below.
 - The *Automatic Scaling* check box uses the limits of the map as the limits for the axes, and uses reasonable tick mark spacing along the axis. When any parameters are changed in the axis properties dialog, the *Automatic Scaling* check box is switched off. After making any changes, you can return to the automatic settings by clicking on the *Automatic Scaling* check box.
 - The *Axis Minimum* box specifies the minimum axis limit.
 - The *Axis Maximum* box specifies the maximum axis limit.
 - The *Major Interval* box specifies the spacing, in data units, between major ticks on the selected axis. Enter the desired spacing to use between major ticks on the axis. See the *Axis Ticks and Tick Labels* section in this chapter for more information on ticks and tick placement on an axis.
4. Click OK or *Apply* to redraw the axis with the new settings.



Scaling

☒ Automatic Scaling

Axis Minimum: Cross Y Axis at:

Axis Maximum: Cross Z Axis at:

Major Interval:

First Major Tick:

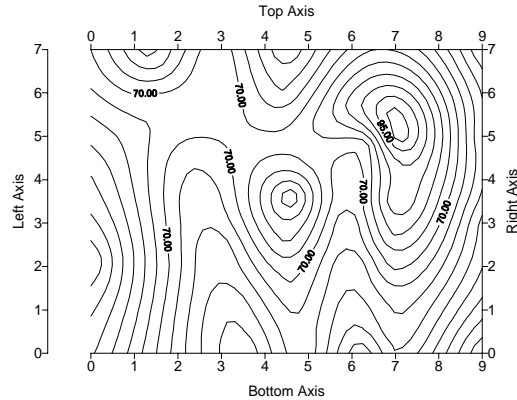
Last Major Tick:

*You can use the **Scaling** page to set the axis minimum, maximum, and interval.*

Positioning Axes

Axes are a part of the map or variogram on which they are placed and by default they are drawn at the edge of a map or variogram.

For example, a contour map has two Y axes, the Left Axis and Right Axis. The placement of the Y axes is relative to the data units along the X axes (the Top Axis and Bottom Axis) on the map. By default, the left Y axis crosses the X axes at the minimum X axis value.



The Left Axis is offset from the map by specifying the position in relation to the Bottom axis. In this example, the Left axis crosses the Bottom Axis at a value of minus one.

Individual axes cannot be moved with the mouse, but you can position axes relative to the map by selecting the **Scaling** page in the axis properties dialog.

To change the position of an axis:

1. Double-click on the axis.
2. Click the **Scaling** page in the axis properties dialog.
3. You can define the positioning of the axes by entering values into the *Cross Axis at* boxes.
 - The *Cross Y Axis at* box is only displayed when an X or Z axis selected. This specifies the position of the X or Z axis along the Y axis. The value is entered in Y data units.
 - The *Cross X Axis at* box is only displayed when a Y or Z axis selected. This specifies the position of the Y or Z axis along the X axis. The value is entered in X data units.
 - The *Cross Z Axis at* box specifies the position of the X or Y axis relative to the Z axis on a wireframe or surface. The value is entered in Z data units. By default, the X and Y axes are drawn at the elevation of the base defined for the wireframe or surface. For example, if the base for the wireframe is set to be 25, the X and Y axes cross the Z axis at an elevation of 25.

Cross Y Axis at:	24
Cross Z Axis at:	0

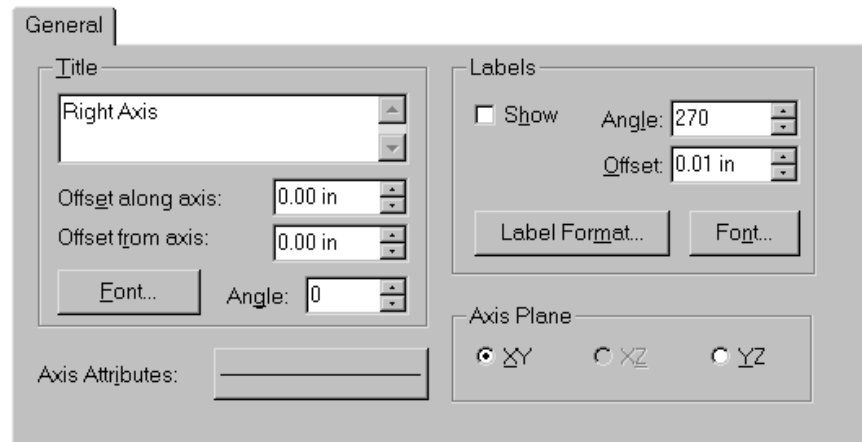
*The Cross Axis at boxes on the **Scaling** page give you control over the position of the axis.*

Axis Titles

Axis titles can use any text properties including math text formatting instructions. See *Appendix B, Math Text Instructions* for more information on text properties and math text instructions.

To place a title on an axis:

1. Double-click on the axis to display the axis properties dialog.
2. Select the **General** page to edit the axis title.
3. Type the axis title into the text box in the *Title* group. Titles can contain multiple lines and can include math text instructions. To create a new line, press CTRL+ENTER.

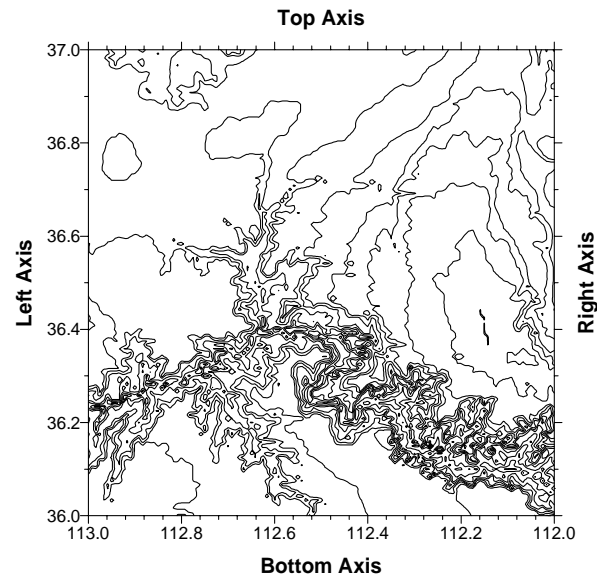


*The title "Right Axis" is entered into the Title box on the **General** page in the axis properties dialog.*

4. Set either of the offset values in page units.
 - The *Offset along axis* box controls the placement of the title along the axis. Positive offset values move the axis title in the positive direction along the axis.
 - The *Offset from axis* box controls the placement of the title towards or away from the axis. Positive offset values move the axis title away from the axis.
5. Click the *Font* button to specify the text properties for the axis title.
6. The *Angle* box controls the angle at which the title is drawn. Positive axis angles rotate the title in a counterclockwise direction.
7. Click OK or *Apply* in the axis properties dialog and the axis is redrawn with a title.

Axis Ticks and Tick Labels

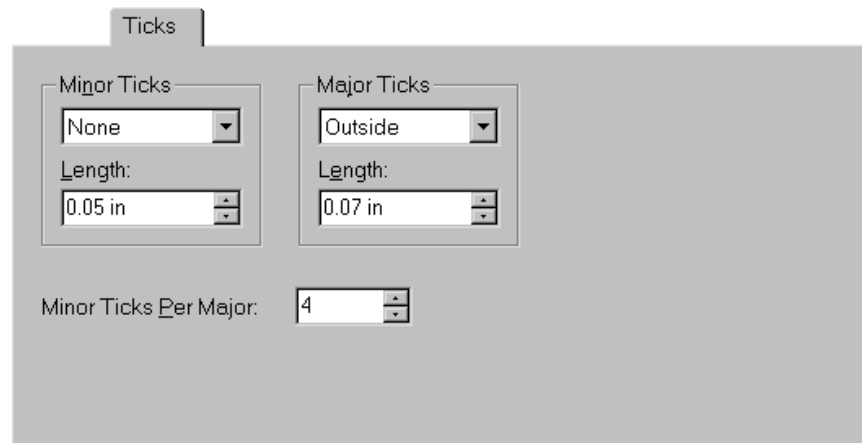
Axes can display both major and minor ticks. Major ticks can display tick labels. Minor ticks are displayed at regular intervals between major ticks and do not have associated labels. The display of tick marks is controlled from the **Ticks** page in the axis properties dialog. The spacing of the major ticks along the axis is controlled from the **Scaling** page in the axis properties dialog. Tick labels are controlled from the **General** page in the axis properties dialog.



This is a contour map showing major tick and minor tick marks.

Major Ticks

When a map is created, major ticks are displayed by default. You can control the visibility, position, and length of major ticks. Major ticks display tick labels along the axis.

The image shows a dialog box titled 'Ticks'. It has two main sections: 'Minor Ticks' and 'Major Ticks'. In the 'Minor Ticks' section, there is a dropdown menu set to 'None' and a 'Length' spinner box set to '0.05 in'. In the 'Major Ticks' section, there is a dropdown menu set to 'Outside' and a 'Length' spinner box set to '0.07 in'. Below these sections, there is a label 'Minor Ticks Per Major:' followed by a spinner box set to '4'.

*You can use the **Ticks** page in the axis properties dialog to control tick properties.*

1. Double-click on the axis to display the axis properties dialog.
2. Select the **Ticks** page.
3. The *Major Ticks* group controls the display and length of major ticks.
 - You can use the major ticks list to specify whether the major ticks are drawn *Outside*, *Inside*, or *Cross* the selected axis. If you do not want to show major ticks, select the *None* option from the list.
 - The *Length* box controls the length of the major ticks. The tick lengths are in page units.
4. Click OK or *Apply* and the map is redrawn with the specified major tick parameters.

Major Tick Spacing

On the **Scaling** page in the axis properties dialog there are three boxes that control the spacing of major ticks along the selected axis: the *First Major Tick*, *Last Major Tick* and *Major Interval* boxes.

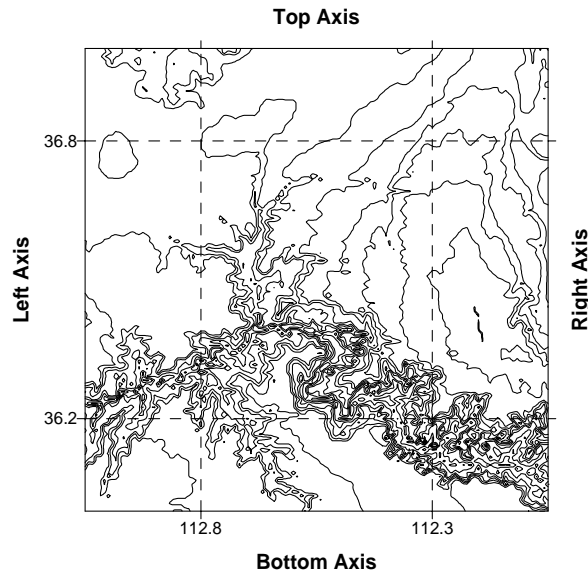
To edit major ticks on an axis:

1. Double-click on the axis.
2. Click the **Scaling** page in the axis properties dialog.
3. You can specify the position for the first and last major ticks, and the spacing in data units between each major tick along the axis.
 - The *First Major Tick* box contains the value of the minimum major tick mark. This value is expressed in data units. The *First Major Tick* does not have to be the same as the axis minimum but it cannot be less than the axis minimum.
 - The *Last Major Tick* box contains the value of the maximum allowed major tick. This value is expressed in data units. The *Last Major Tick* does not have to be the same as the axis maximum but it cannot be greater than the axis maximum.
 - The *Major Interval* box contains the value of the spacing, in data units, between major ticks along the selected axis. The major tick marks occur at values

$$\text{Tick} = \text{First Major Tick} + n \times \text{Major Interval} \quad n=0,1,\dots,N$$

where N is the largest integer such that $\text{Tick} \leq \text{Last Major Tick}$. Thus, a tick mark is placed at the value of the *Last Major Tick* only if the interval ($\text{Last Major Tick} - \text{First Major Tick}$) is evenly divisible by *Major Interval*.

4. Click OK or *Apply* and the axis is redrawn with the selected tick parameters.



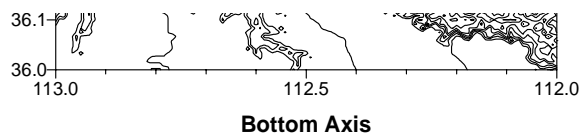
By setting the First Major Tick, Last Major Tick and Major Interval values, you can control the position of major ticks along the axis.

Minor Ticks

Minor ticks are defined as the number of ticks drawn between adjacent major ticks. Minor ticks do not have tick labels and are not drawn on an axis by default.

To display minor ticks on an axis:

1. Double-click on the axis.
2. Click the **Ticks** page in the axis properties dialog.
3. On the **Ticks** page, the *Minor Ticks* group controls the display and length of minor ticks.
 - The minor ticks list specifies whether the minor ticks are drawn *Outside*, *Inside*, or *Cross* the selected axis. If you do not want to show minor ticks, select the *None* option from the list.
 - The *Length* box controls the length of the minor ticks. The tick lengths are in page units.
4. To specify the number of minor ticks to draw between adjacent major ticks, enter the number into the *Minor Ticks per Major* box. This should be one less than the number of minor divisions you want to appear between adjacent major ticks. For example, if you want the minor ticks to indicate 10 divisions, the *Minor Ticks per Major* value should be set to 9.
5. Click OK or *Apply* and the axis is redrawn with minor ticks.



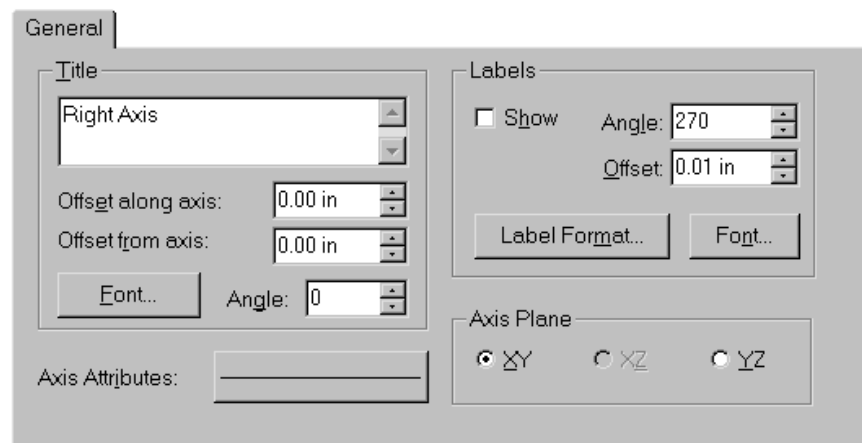
On this axis, four minor ticks are included with every major tick.

Tick Labels

Tick labels are the numbers or text that appear at major tick mark locations. Tick labels are controlled from the **General** page in the axis properties dialog. Tick labels indicate the axis values in data units.

To edit tick labels:

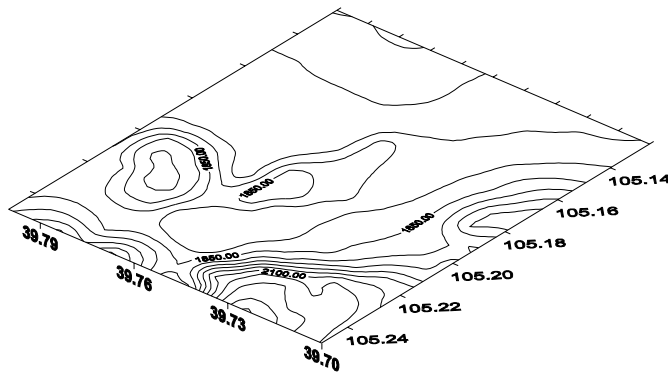
1. Double-click on the axis to display the axis properties dialog.
2. On the **General** page, the *Labels* group controls the display of tick labels.



The General page provides control over the tick labels.

- When the *Show* check box is activated, all major ticks on the selected axis are displayed with a label.
- The *Angle* box specifies the angle at which the tick labels are drawn. Positive angles rotate each axis tick label in a counterclockwise direction.
- The *Offset* box controls how far away the tick labels are drawn from the ticks. This value is in page units.
- Click the *Label Format* button to display the **Label Format** dialog, in which you specify the numeric format used for the tick labels. Refer to the next section for more information on label formats.
- Click the *Font* button to display the **Text Properties** dialog and set the text properties.

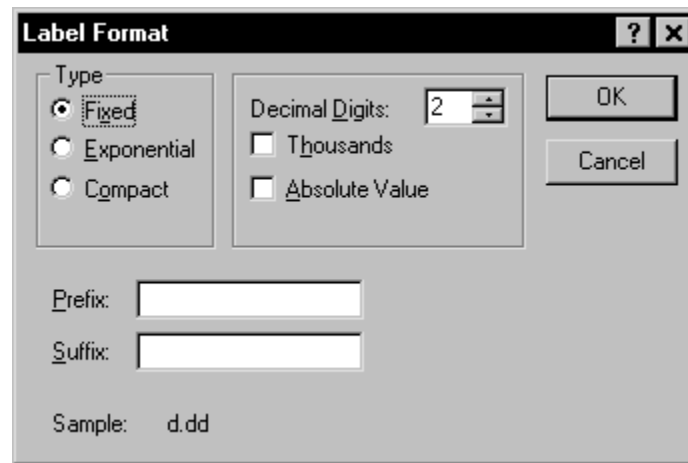
3. You can use the *Axis Plane* group to choose the plane in which the tick labels are drawn. For maps with a 90-degree tilt, axis tick labels must appear in the XY plane to be seen.
 - The *XY* option draws the tick labels in the plane defined by the X and Y axes. Use this setting for planar view maps such as contour maps.
 - The *XZ* option draws the tick labels in the plane defined by the X and Z axes. This option is used only when the tilt for the map is less than 90 degrees.
 - The *YZ* option draws the tick labels in the plane defined by the Y and Z axes. This option is used only when the tilt for the map is less than 90 degrees.
4. Click *OK* or *Apply* and the axis is redrawn with the selected tick labels.



Axis tick labels can be drawn in different axis planes, and can be rotated to make reading easier.

Tick Label Format

The **Label Format** dialog allows you to specify the numeric format for the axis tick labels. For example, you might want to change the number of digits after the decimal point on the tick labels. The **Label Format** dialog allows you to change this parameter, as well as others controlling the label format.



*You can use the **Label Format** dialog to choose the tick label format.*

To change the tick label format:

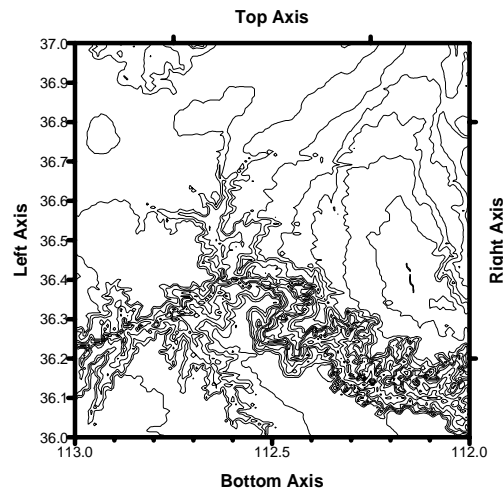
1. Double-click on the axis to display the axis properties dialog.
2. Click on the **General** page.
3. Click the *Label Format* button to display the **Label Format** dialog. As you change the parameters for the labels, a sample of the label appears in the lower left corner of the dialog.
 - The *Fixed* option displays numbers as dd.dd where dd represents one or more digits.
 - The *Exponential* option displays numbers as d.ddE+ddd where d is a single digit, dd can be one or more digits, and +dd is a positive or negative sign followed by three digits.
 - The *Compact* option displays the labels as fixed or exponential, whichever requires fewer digits.
 - Use the *Decimal Digits* box to set the number of digits to the right of the decimal point with *Fixed* and *Exponential* types. When *Compact* is selected, enter the number of *Significant Digits*.
 - The *Thousands* check box displays numbers with a comma between every three digits to the left of the decimal point.
 - The *Absolute Value* option displays the absolute value for all labels. Negative signs are not displayed when this check box is activated.
4. If you want to add a prefix or suffix to all the tick labels on the axis, enter the text into the *Prefix* or *Suffix* boxes.
5. Click OK to return to the axis properties dialog.
6. Click OK or *Apply* in the axis properties dialog to redraw the axis with the new labels.

Axis Tick and Line Properties

The axis ticks and the axis line share the same line properties.

To set the line properties:

1. Double-click on the axis to display the axis properties dialog.
2. Select the **General** page.
3. Click the *Axis Attributes* button to display the **Line Properties** dialog.
4. Choose the line style, color, and width for the selected axis and then click OK.
5. Click OK or *Apply* in the axis properties dialog and the axis and axis ticks are drawn with the assigned line properties.

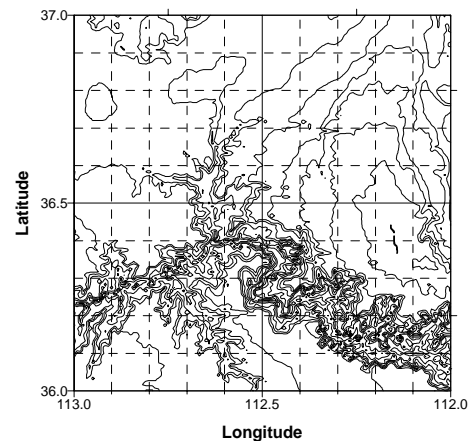


In this map, the axes use a heavier weight line than the default line width.

Grid Lines

Grid lines are a mesh over the top of the map. Along an axis, grid lines can originate from the major ticks, minor ticks, or both. The major and minor grid lines can use different line properties. When grid lines are added to wireframes or surfaces, the lines are drawn on the plane defined by the base of the map.

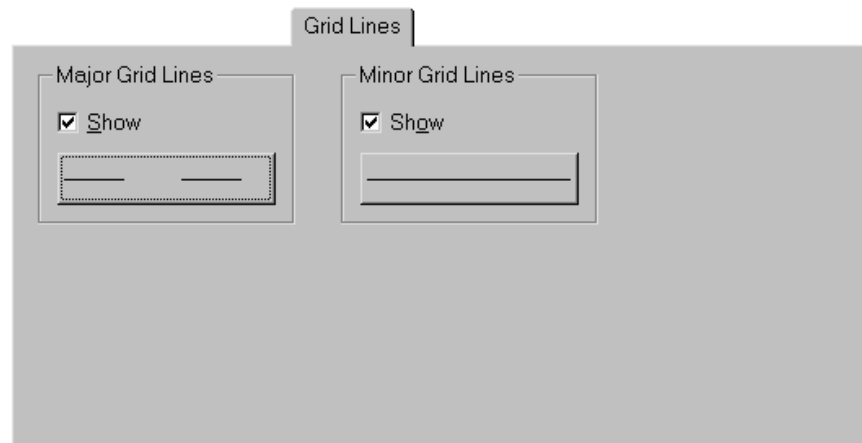
Grid lines are drawn over the extent of the map. For example, if you draw grid lines on an X axis, the grid lines are drawn parallel to the Y axis over the entire map. If your X axis is offset from the Y axis, the grid lines are not drawn beginning at the end of the X axis, but begin at the minimum Y value on the map.



Grid lines originate from the tick marks. Different line properties can be used for the major and minor grid lines.

To draw grid lines:

1. Double-click on an axis to open the axis properties dialog.
2. Specify the tick spacing to use for the grid lines on the **Scaling** page. Grid lines can be based on both major and minor ticks on the axis. The ticks do not need to be displayed for the grid lines to be drawn.
3. Click the **Grid Lines** page.
4. If you want grid lines to extend from the major ticks, click the *Show* check box in the *Major Grid Lines* group.
5. Click the line properties button to set the line properties for the grid lines.
6. Use the *Minor Grid Lines* group to control the display and properties of grid lines originating from minor ticks.
7. Click OK or *Apply* and the grid lines are drawn on the map or variogram.



Check the Show boxes to display grid lines based on the major and minor ticks.

Chapter 16

Adding Color to Maps

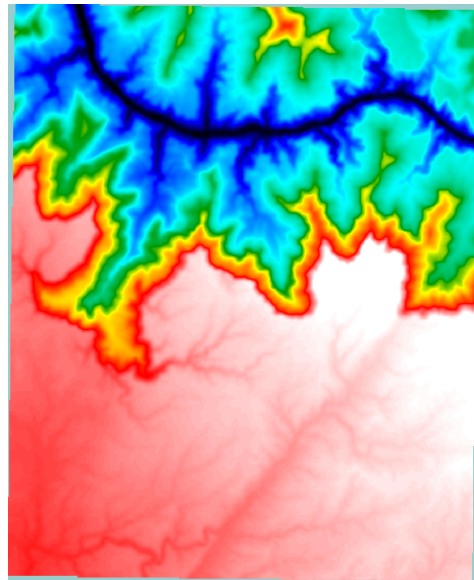
Introduction to Color Spectrums

Color fill can be added to several map types and objects in **Surfer**. Contour maps can be line drawings or they can have color fill. The vector symbols on a vector map can be colored. You can choose a preset color spectrum for image maps, shaded relief maps, contour maps, and surfaces or you can create your own color spectrum.

Colors are assigned through the **Fill, Fill Properties**, or **Color Spectrum** dialogs, depending on the type of object. In some cases, color spectrums can be saved for future use as color spectrum files [.CLR] or as level files [.LVL]. See *Appendix C, File Formats* for more information on color and level file formats.

Colors

Contour maps, image maps, shaded relief maps, vector maps, and surfaces all use the **Color Spectrum** dialog to assign colors to the map. The color spectrum has specific colors assigned to nodes along the spectrum. **Surfer** automatically blends colors between the nodes to produce a smooth color gradation over your map. After you create a color spectrum, you can save the spectrum for later use with other maps. Although the colors are assigned to specific values on a map, the spectrum can be used with maps containing different Z ranges because the node values are stored as percentages.



Color spectrums, such as in the image map above, can be saved for future use in other maps.

Opening the Color Spectrum Dialog

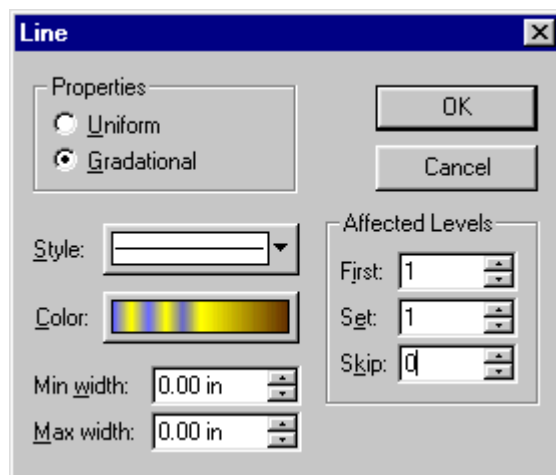
The **Color Spectrum** dialog is opened slightly differently for each map type.

Contour Maps

Contour map fill colors are opened by clicking the *Fill* button on the **Levels** page in the contour map properties dialog. Contour map lines also use the **Color Spectrum** dialog if the gradational line color option is selected.

To select the gradational line option,

1. open the contour map properties dialog,
2. select the **Levels** page,
3. click the *Line* button,
4. select the *Gradational* option,
5. and then click the *Color* button.




*Click on the Line button on the **Levels** page in the contour map properties dialog to create gradationally colored lines on a contour map.*

Image and Shaded Relief Maps

To open the **Color Spectrum** dialog for an image or shaded relief map, open the map properties and then click the *Colors* button on the **General** page.

Vector Maps

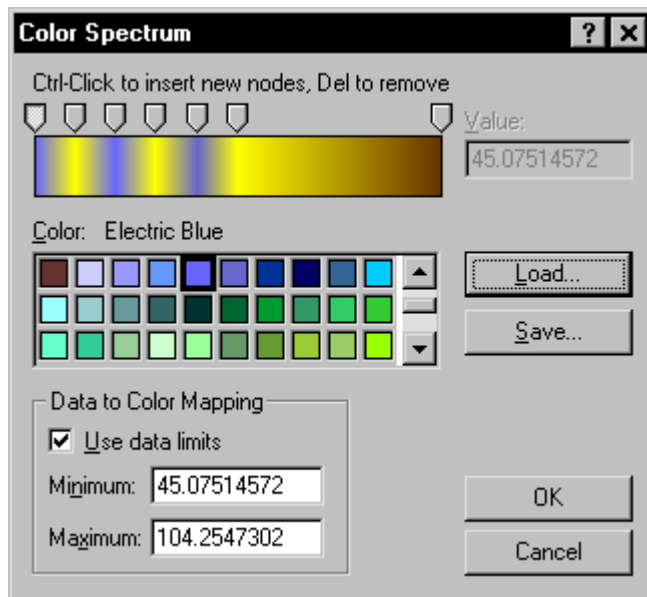
The *Color Scaling* group on the **Symbol** page in the vector map properties dialog contains options for gradationally coloring the symbols by magnitude or by a grid file. If you select the grid file option, choose a grid file by clicking the  button to the right of the file name box. You must use a grid that has identical grid line geometry to the vector map grid file (the same number of rows and columns). Click the *Vector Colors* button to open the **Color Spectrum** dialog.

Surface

To open the **Color Spectrum** dialog for a surface, open the map properties and then click the *Upper* button in the *Material Color* group on the **General** page.


The Color Spectrum Dialog

The **Color Spectrum** dialog allows you to specify the colors to associate with a range of data within the map. Blended colors are achieved by creating anchor points (nodes) along a color bar and assigning specific colors to the specified anchor points. Colors are linearly interpolated between adjacent anchors.



Use the **Color Spectrum** dialog to define the map colors.

Anchor Nodes

Anchor nodes are represented by a slider button  above the color spectrum. The data value for the node is displayed in the *Value* box. You can add additional anchor nodes at any position along the color spectrum. To create a new anchor node, hold down the CTRL key on your keyboard and click above the color spectrum. The new anchor is automatically assigned the current color in the color palette.

You can change colors by clicking once on the anchor slider button and then clicking on any color in the color palette. The color spectrum is updated to show the change.

Chapter 16 - Adding Color to Maps

To position an anchor node:

1. Move the mouse pointer over the slider button.
2. Click and hold the left mouse button.
3. Drag the slider button to the desired position and release the left mouse button. As you move the slider button, the data value is indicated in the *Value* box. Alternatively, you can set the anchor to a specific data value by entering a number into the *Value* box. Note that the beginning and ending anchors cannot be moved or deleted.
4. You can reset the *Minimum* and *Maximum* to change the range, and therefore, the position of the starting and ending nodes. For more information on ranges, see the *Using Color Spectrums in Map Series* section of this chapter.

To select a color to associate with the anchor node:

1. Click on the slider button you wish to modify.
2. Click on the desired color in the color palette. The color spectrum is updated to indicate the change.

You can add as many anchor nodes as you want. This lets you blend colors in many different ways on the color spectrum. You can delete anchor nodes from the color spectrum by clicking the slider button and then pressing the DELETE button on your keyboard.

Loading a Preset Color Spectrum

The *Load* button opens an existing color spectrum file [.CLR]. When you click the *Load* button, the **Open** dialog is displayed with a list of color spectrum files. Double-click the file you want to use with the current map, and the color spectrum is updated to show the color spectrum file settings. Several predefined color spectrum files are supplied with **Surfer** in the SAMPLES directory.

Color Spectrum Data Range

The *Data to Color Mapping* group contains options for setting the minimum and maximum data values to use in the color spectrum. Check the *Use data limits* check box to use the data minimum and maximum values. If you would prefer to set the color spectrum minimum and maximum values to other values, enter the new numbers into the *Minimum* and *Maximum* boxes. This is useful when you are mapping different data sets in a similar range and would like to have the same data values represented by the same colors each time. If a data value within the map falls outside this range, it is assigned the minimum or maximum color, whichever is closest.

Saving a Color Spectrum

Click the *Save* button in the **Color Spectrum** dialog to create a color spectrum file [.CLR] based on the current color spectrum settings. Once you have typed a file name in the **Save As** dialog, click the *Save* button to create a new [.CLR] file. This file can then be used with other data files. Refer to *Appendix C, File Formats* for more information on the color spectrum file format.

Reusing a Color Spectrum

Once you have saved a color spectrum file, you can use it in additional maps by clicking the *Load* button in the **Color Spectrum** dialog. Color spectrum files can be used with maps of varying *Z* ranges since the anchor nodes are stored as percentages rather than as data values. To use the exact same colors in the exact same data locations when the *Z* ranges vary slightly, please refer to the *Using Color Spectrums in Map Series* section of this chapter.

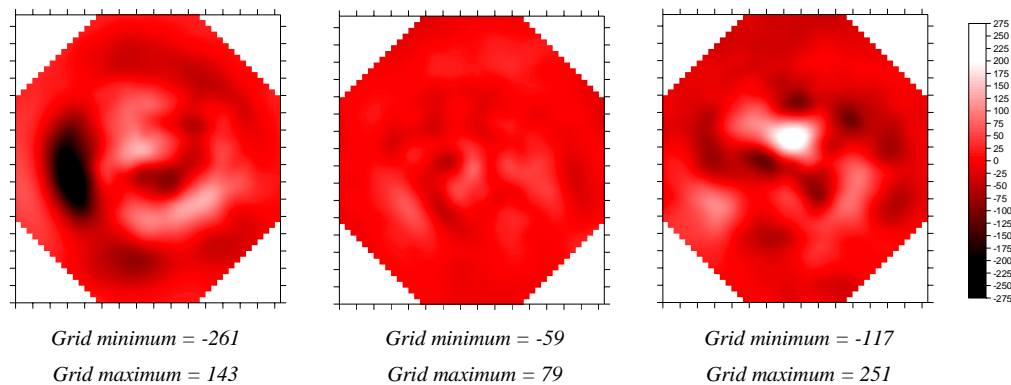
Using Color Spectrums in Map Series

Sometimes it is necessary to use the same range of colors on different maps. By default, color spectrums are assigned to the minimum and maximum data values for a given data set. If the maps have different data ranges, the colors will not map to the same data values and it will be difficult to compare the maps. In this case, you may want to override the default data limits and assign custom values in the *Data to Color Mapping* group.

The following example contains steps on how to create maps in a series that contain the same color spectrums although the data minimums and maximums differ. The example image maps show the same gridded area at three different points in time. To accurately compare the maps, the color spectrum needs to be the same through time even when the range of values fluctuates between grids.

Know Your Data

The first step in producing maps with the same color series is to know your data. Find the minimum and maximum data values for all of your data sets before you begin. This will help you to decide the minimum and maximum values to use in the color spectrum for the series of maps. The maximum Z value is +251 in one of the grids used in this example, and the minimum Z value is -261. The largest and smallest Z values do not appear in the same grid.



The three maps shown above all use the same color spectrum range, even though the grid minimums and maximums differ.

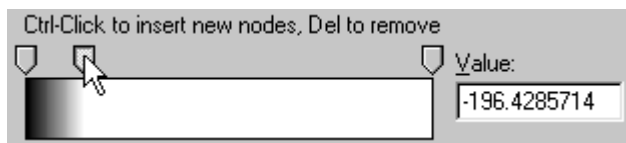
The three maps above show the same phenomena over a period of time. The grid ranges are not the same, but since the maps are to be compared, the colors need to map to the same data values for all maps. Using the same color spectrum file and changing the minimum and maximum spectrum data values allows you to compare the maps based on the same color spectrum. The color scale bar to the right is used for all three maps. The map on the left contains a dark black area because it has some extreme negative values, but it does not have any pure white because it does not have extreme positive values. The map on the right is opposite, it contains extreme positive values so there is an area of pure white, but the negative values are not extreme so there is no pure black. The map in the center does not contain extremes in either direction so it is mostly gray. Note that these maps have "stair step" edges due to blanking in the corners. A smaller grid spacing or interpolating the pixels would smooth the edges.

Creating a Color Spectrum File

After you know the series minimum and maximum and have decided on a spectrum minimum and maximum, you can create a color spectrum file to use with all of the data. In this example, -275 is the minimum and +275 is the maximum.

To create a color spectrum file:

1. Create a map from one of the grid files. An image map is used throughout this example, though you can use contour maps, image maps, shaded relief maps, vector maps, or surfaces in this procedure.
2. Double-click on the map to open the map properties.
3. Open the **Color Spectrum** dialog.
4. In the *Data to Color Mapping* group, change the *Minimum* to -275 and the *Maximum* to 275.
5. Hold the CTRL key down on your keyboard and click on the negative end of the color spectrum to add an anchor node. While the node is still selected, change the *Value* to -200 or slide the node to the -200 position (watch the numbers in the *Value* box). The minimum color (at -275) is black. Select black for the -200 node as well.



*Hold down the CTRL key and click on top of the color spectrum to add a node.
To position a node, slide the node or enter a number into the Value box.*

6. Repeat step five to add nodes at zero (gray) and +200 (white).
7. Click the *Save* button to save the color spectrum file.

Using the Color Spectrum File in Other Maps

After you have created the color spectrum file as above, you can use it in the remainder of the maps in a series.

To use a color spectrum file in another map:

1. Create the map. In this example, a new image map is created with the next grid file.
2. Open the map properties by double clicking on the map name in the **Object Manager**.
3. Click the *Colors* button in the **General** page in the image map properties dialog.
4. Click the *Load* button in the **Color Spectrum** dialog and open the color spectrum file you just created.
5. In the *Data to Color Mapping* group, set the *Minimum* and *Maximum* to the pre-determined values. In this example, the minimum is set to -275 and the maximum to 275.
6. Repeat this procedure for the rest of the maps in the series.

Level Files

Contour Maps

For contour maps, level files [.LVL] contain information about contour levels, including the line properties, color fill, label frequency, and hachure information. The level file contains all the information contained on the **Levels** page in the contour map properties dialog. After defining custom contour levels and colors on a map, you can save the level and color information in a level file. A level file can be recalled for any other contour map or wireframe. Not all of the information in a level file can be used in a wireframe. If you use a level file created in the contour map properties dialog in a wireframe, the color fill, contour label, and hachure information are ignored. For information on the format of level files, see *Appendix C, File Formats*.

Wireframes

For wireframes, level files contain information for the Z level line properties. After defining custom Z levels and line properties on the **Z Levels** page, you can save the level and color information in a level file. This level file can be recalled for any other wireframe or contour map.

Level files created in the wireframe properties dialog contain only information for the Z level line properties. A level file created from the wireframe properties dialog can be used for a contour map and vice versa. Any additional information not used is ignored.

Creating Level Files

The easiest way to create a level file is from the contour map properties or wireframe properties dialogs. You can also create a level file in the worksheet or in an ASCII editor, using the file format in *Appendix C, File Formats*.

To create a level file:

1. In the map properties dialog, click on the **Levels** page (contour) **Z Levels** page (wireframe), or **Color Zones** page (wireframe).
2. Change any of the contour line parameters, including the contour levels, line properties, fill properties, contour labels, and hachure information. On wireframes, you can only change the contour levels and line properties.
3. After changing the settings, click the *Save* button.
4. In the **Save As** dialog, type the name of the level file and click *Save*. The level file is saved with an [.LVL] extension.

To use this file with any contour map or wireframe, click the *Load* button on the **Levels** page, **Z Levels** page, or **Color Zones** page. All maps using the level file must have comparable Z data ranges; otherwise, the map will appear empty.

Chapter 17

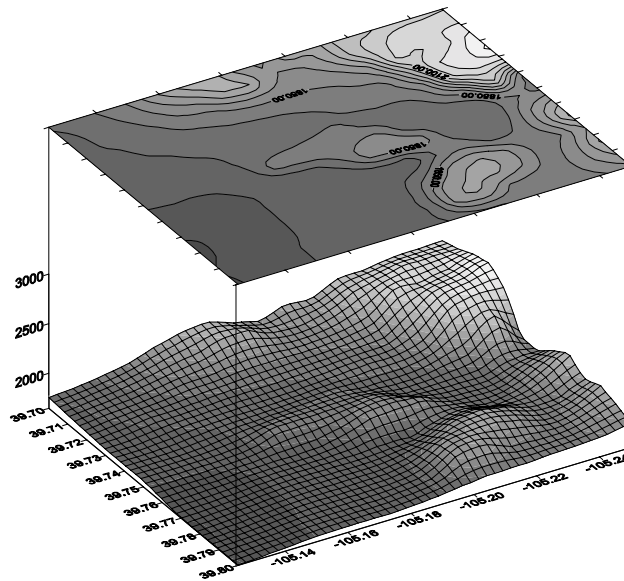
Positioning and Overlaying Maps

Introduction to Positioning and Overlaying Maps

There are several ways to arrange one or more maps on a page in **Surfer**. The simplest technique is to simply drag the map to the desired position with the mouse, just like any other drawing object in **Surfer**. Multiple maps can be independently positioned on a single page, or overlaid into a single composite map. Maps can also be stacked above one another in order to align the coordinate systems horizontally.

The **Map | Overlay Maps** command overlays all selected maps. Most combinations of map types can be overlaid. The exceptions are combining wireframes and surfaces, overlaying raster maps with wireframes, and overlaying multiple wireframes. Raster maps include shaded relief maps, image maps, surfaces, and base maps containing a bitmap. When you overlay maps, **Surfer** determines the map limits and correctly superimposes the map coordinate systems. Some examples include overlaying a filled contour map on a wireframe, or draping a bitmap over a surface.

The **Map | Stack Maps** command horizontally aligns the map coordinate systems on the page. This is useful for stacking related maps above each other, such as a contour map over a wireframe. The **Stack Maps** command is designed to align two maps using the same grid limits and oriented with the same view parameters.



*Surfer can combine maps into a single composite map and stack maps over one another. In the example here, a filled contour map is overlaid on a wireframe on the bottom of the figure, and a contour map is stacked above it. **Draw | PolyLine** was used for the lines connecting the wireframe and contour maps.*

Moving Maps

The easiest method of moving maps on a page is to move the map with the mouse.

To move a map:

1. Select the map and the selection handles appear surrounding the map.
2. Move the pointer within the limits indicated by the selection handles.
3. Press and hold the left mouse button and drag the map to any position on the page. As you drag the map, the extents of the map are indicated by a dashed rectangle.
4. When the dashed rectangle is in the desired position on the page, release the left mouse button and the map is drawn in the new position.

Placing Several Maps in the Same Plot Window

Any number of maps can be created within a single plot window. As each map is created, it is initially placed in the center of the page. You can drag each map to the desired position on the page after it is created. In addition to creating new maps in a single plot window, existing maps can be added to a page by pasting them into the plot window. Each of these methods is discussed below.

Creating Several Maps in the Same Plot Window





When you create several maps in the same plot window, the default map size often results in overlapping maps even after they are carefully placed on the page. If this is the case, see *Chapter 14, Common Map Properties* for more information on how to size maps.

To create several maps in the same plot window:

1. In an empty plot window, create a map using the commands available in the **Map** menu.
2. If the default scaling is not satisfactory, use the map property **Scale** page to adjust the map size.
3. Select each map and drag it to the position on the page where you want it to appear.
4. Repeat the previous steps as many times as needed.

Placing Several Completed Maps in the Same Plot Window

If you have already created and saved the maps you want to display together, they can be combined into one plot window.

1. Open each map using **File | Open** or click the  button. This automatically opens a new plot window for each map.
2. Choose **File | New** or click the  button to create a new empty plot window. This window will contain all the maps to be displayed.
3. Open the window containing one of the maps by choosing the name of the plot window from the **Window** menu.
4. Select the map.
5. Choose **Edit | Copy** or click the  button.
6. Select the window to receive all the maps using the **Window** menu.
7. Choose **Edit | Paste** or click the  button. The map is pasted into the plot window. These maps can be edited.
8. Repeat steps 3 through 6 for each map to be placed in the plot window.

Aligning Several Maps on the Same Page

Surfer provides several ways to align maps on the same page.

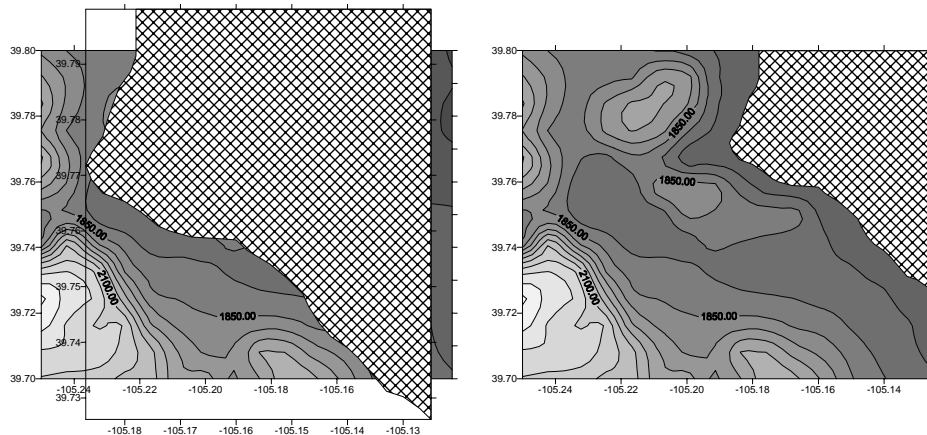
- Use the mouse to manually drag the maps on the page. This method is rather imprecise, but provides complete control over the map position. The arrow keys can be used to "nudge" the selected map one pixel at a time for more precise positioning.
- Use **Map | Stack Maps** to align maps placed one above the other on the page. Maps are moved horizontally on the page so their coordinate systems and origins are aligned. See the *Stacking Maps* section of this chapter for more information.
- Use **Arrange | Align Objects** to align selected objects relative to the bounding box surrounding the objects. The objects may be aligned both vertically and horizontally. Map alignment done in this way is based on the map's bounding box and not on the coordinate system for the maps. The bounding box for a map is indicated by the selection handles. See *Chapter 21, Creating, Selecting, and Editing Objects* for more information.
- Use the **Rulers and Grid** page in the **File | Preferences** command to select the *Snap to Ruler* option. When the *Snap to Ruler* option is on, the upper left corner of the object's bounding box is snapped to the nearest ruler division when the object is moved. See *Chapter 23, Preferences* for more information on the *Snap to Ruler* option.

Overlaying Maps

It is possible to combine several maps created from related data by overlaying maps. You can overlay any combination of base maps, contour maps, vector maps, post maps, image maps, shaded relief maps, or surfaces. Maps can contain only one wireframe overlay, however. Other exceptions are noted on page 373.

When you overlay maps, the overlays use a single set of X, Y, and Z axes and are positioned according to the composite coordinate system. If two or more overlays use the same limits, they will overlay on top of one another. If the overlays cover adjacent areas, they are placed in the correct position relative to one another and a single set of axes span the entire range in the composite map. Composite maps become a single object and are moved and scaled as a single entity.

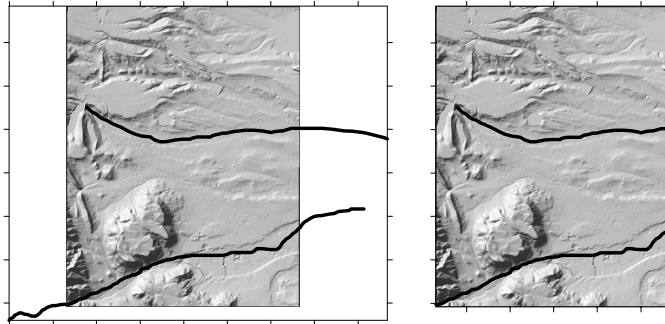
Consider a contour map and a base map that displays the outline of a lake on the contour map. The limits of the base map are the XY extents of the lake and are not the same as the contour map limits. If you create both the base map and the contour map in a single plot window, they do not overlay correctly because by default the maps are scaled differently. In addition, each map uses a different set of XY axes. If you select both maps and then choose **Map | Overlay Maps**, the contour map and the base map are rescaled and combined into a single map, or composite map, using a single set of axes. The lake is now correctly positioned on the contour map.



Before you overlay maps, the two maps might use different map limits, and each map uses its own axes. When you overlay the maps, they are positioned properly with respect to each other, and use only one set of axes.

Overlay Limits

Overlays containing different limits can be clipped to any limit. For example, you might overlay a base map on a shaded relief map. If the base map has greater extent than the shaded relief map, the axes are set at the limits of the base map. You can clip the limits for the composite map by using the map properties **Limits** page. In this case, you can set the limits to the shaded relief map limits. For more information on limits, see *Chapter 14, Common Map Properties*.



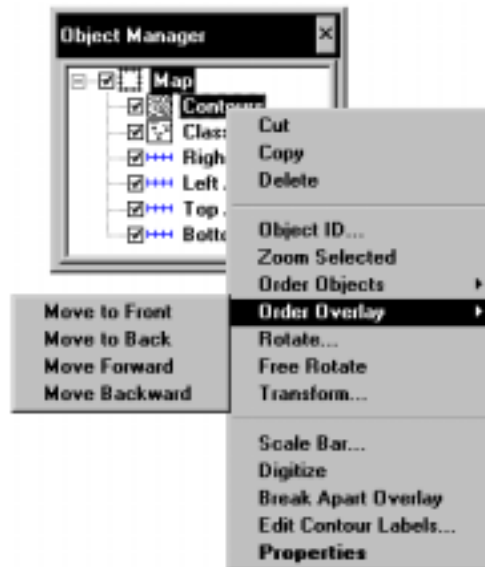
*Overlays can be clipped to the limits of other overlays in a composite map by selecting the **Limits** page in the map properties dialog.*

Overlay Order

When overlaying maps, the top map may obscure the underlying maps. For example, you might overlay a filled contour map on a post map. If the contour map is on top, the post map is not visible. You can change the order of the maps through the **Object Manager**.

To change the order of the maps:

1. In the **Object Manager**, click on the overlay to be displayed on top.
2. Right-click on the map name.
3. Select **Order Overlay**.
4. Select **Move to Front** and the map is redrawn with the specified overlay on top.



*If an overlay is obscured, use **Order Overlay** in the **Object Manager** to rearrange the overlays.*

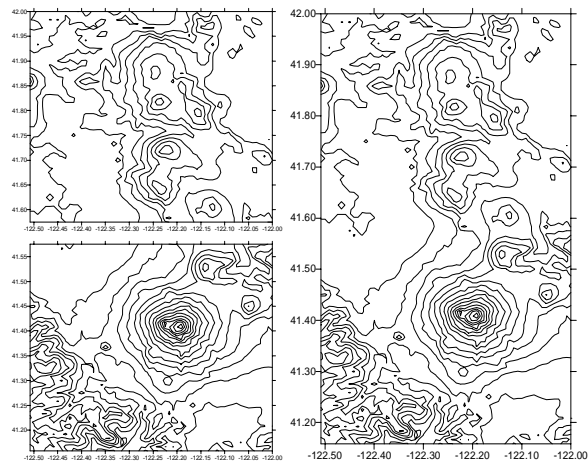
When you overlay maps on wireframes or surfaces, the maps are draped over the surface. The wireframe or surface is drawn in the usual fashion but the overlaid maps are "draped" over the top.

Wireframes and Hidden Lines

Hidden lines are not removed from wireframes. For example, contour lines are not hidden when the contour map is overlaid on a wireframe. If this is a problem, consider experimenting with surfaces and overlays instead.

Adjacent Maps

If you have files of adjacent areas, use **Map | Overlay Maps** to precisely position the maps relative to each other. For example, you might have base maps for two adjacent quadrangles. You can create base maps of the two quadrangles in one plot window, select both base maps, choose the **Map | Overlay Maps** command, and then the two maps are drawn side-by-side on the page.



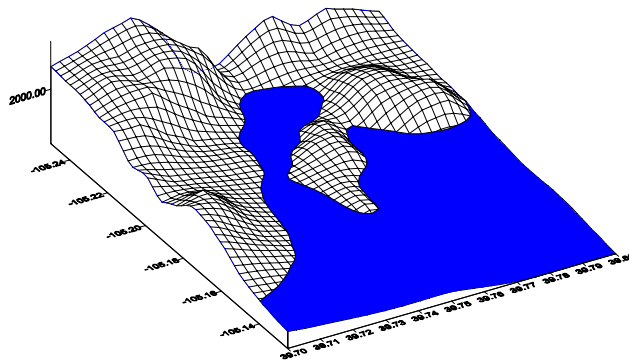
Map | Overlay Maps can combine adjacent maps into a single composite map. In this example, grid files containing elevation data from two adjacent quadrangles are used to produce the maps on the left. Map | Overlay Maps combines the two maps into the single composite map on the right.

If you have multiple grid files that are adjacent to each other, use **Grid | Mosaic** to merge them into one grid. See *Chapter 18, Grid Operations* for more information on mosaic.

Overlaying Maps

Multiple maps can be overlaid if all overlays contain similar XY ranges.

1. Create the maps in a single plot window by choosing the appropriate commands from the **Map** menu. When maps are first drawn in the plot window, they may or may not overlay directly, and each map is an independent object with its own set of axes.
2. Select the maps you want to combine. When selecting multiple objects you can:
 - a. Hold down the SHIFT key and click each map to be combined.
 - b. Press CTRL+A key to select all the maps in the plot window.
 - c. Drag the cursor around the maps to be combined to select them.
3. Choose **Map | Overlay Maps** to combine all the selected maps. The maps are combined in the correct position based on their coordinates and limits.



This is a base map overlaid with a wireframe.

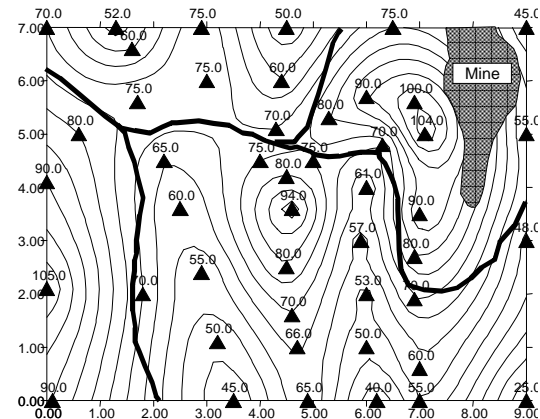
Chapter 17 - Positioning and Overlaying Maps

Editing Overlays

To edit individual overlays in a composite map, open the map properties. Refer to *Chapter 21, Creating, Selecting, and Editing Objects* for more information on selecting objects.

There are several ways to open the map properties:

- select the **Edit | Properties** command when the map is selected
- double-click on the map in the plot window or in the **Object Manager**
- right-click on the map in the plot window or in the **Object Manager** and select the **Properties** command.



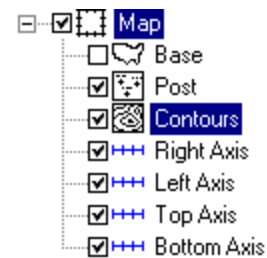
Three maps are combined in this example: a contour map, a post map, and a base map.

Make the desired changes in the **Properties** dialog. Click OK and the overlay is redrawn with the specified changes.

Hiding Overlays

After combining maps, it is possible to hide one or more of the overlays. To temporarily hide a map, uncheck the visibility box next to the map name in the **Object Manager**. The map is redrawn without the selected overlay. To make the overlay visible again, recheck the visibility box.

Note that if a surface is made invisible, the overlays are also made invisible.



Uncheck the box next to the overlay name to hide it. In this example, the base map is hidden.

Deleting Overlays

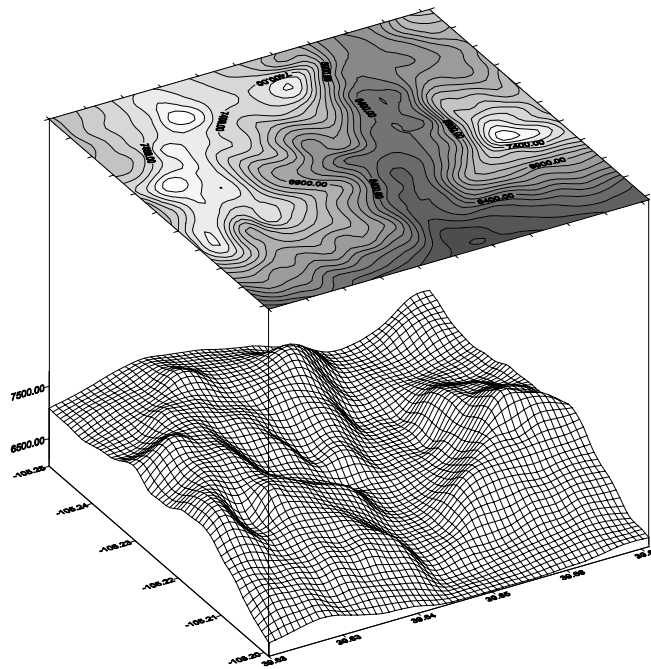
To permanently remove an overlay from a map:

1. Select the overlay using one of the techniques described in *Chapter 21, Creating, Selecting, and Editing Objects*.
2. Use the **Map | Break Apart Overlay** command to break the selected overlay out of the map.
3. Select the new map and press the DELETE key.

Stacking Maps

The **Map | Stack Maps** command aligns individual map coordinate systems horizontally on the page. The **Stack Maps** command was designed to stack maps using similar coordinate systems. This command is useful for keeping two or more maps separated on the page. The vertical position of individual maps on the page is not changed by this command.

When using **Map | Stack Maps** the same view should be used for each map. This insures that points on the stacked maps are aligned properly. If stacking a contour map over a wireframe, select both maps, select **Edit | Properties**, click on the **View** page and then set the view. When you click **OK** or **Apply**, the same view parameters are applied to both maps.



*This is a filled contour map stacked over a wireframe of the same grid file. To connect the two maps, vertical polylines were added with **Draw | PolyLine**.*

The relative vertical placement of maps on the page defines how far apart the maps are after they are stacked. Before using the **Stack Maps** command, position the maps in the approximate location on the page where you want each map to be drawn. The **Stack Maps** command only aligns maps horizontally on the page, not vertically. Stacking maps with perspective projections can be problematic since the Z axis is not vertical on the page. It is better to use the orthographic projection when stacking maps with different Z ranges.

Chapter 18

Grid Operations

Introduction to Grid Operations

The **Grid** menu provides many tools for modifying grid files. **Grid | Spline Smooth** produces grids that result in more rounded shapes on maps. **Grid | Filter** applies methods of digital image analysis to grids, including a broad suite of smoothing (low-pass) filters, contrast enhancement filters, edge enhancement filters, edge detection filters, and general high-pass filters. **Grid | Blank** removes data from specified regions of a grid. Contours are not displayed in blanked regions, and wireframes display blanked regions as low flat regions. **Grid | Convert** changes grid file formats, **Grid | Extract** produces subsets of grids, **Grid | Mosaic** merges grid files, and **Grid | Transform** performs a mathematical transformation on the XY coordinates of a grid. The **Grid | Grid Node Editor** allows you to modify the Z values of individual grid nodes, providing a way to "fine tune" a grid file. The **Grid | Math** command performs mathematical transformations on one or two grid files. The **Grid | Calculus** command calculates derivatives, curvatures, slopes and aspects, differentials, integrals, and spectral analysis.

Any operation used to modify grid files can also be performed on USGS Digital Elevation Models (DEM) or Digital Terrain Elevation Data (DTED). **Surfer** saves the results of these operations in a **Surfer** grid file or ASCII XYZ file format. It does not create, modify, save, or export DEM or DTED files directly.

Spline Smoothing

Spline smoothing uses cubic spline interpolation to compute new grid nodes. The interpolation simulates a drafting technique where a flexible strip (a spline) is used to draw a smooth curve between data points. Spline smoothing does not extrapolate beyond the edge of the original grid file. The original grid limits are used to define the grid limits for the smoothed grid.

A spline is really nothing more than the graphs of a set of contiguous (end-point adjacent) cubic polynomials with the same slopes at their endpoints. Cubic spline smoothing may increase the maximum Z value and decrease the minimum Z value from the input grid.

There are two ways to perform spline smoothing, by expanding the grid or by recalculating the grid. When you expand the grid, nodes are inserted between existing nodes in the original grid.

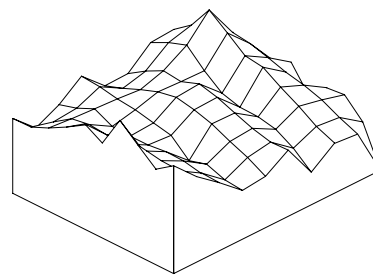
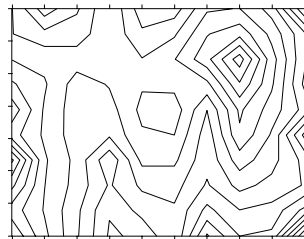
Chapter 18 - Grid Operations

The original grid node values and grid limits are preserved in the smoothed grid, and the new grid nodes are calculated so the smoothest possible contour map or wireframe can be drawn.

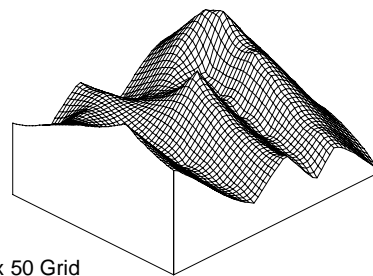
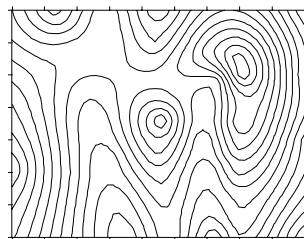
When the grid is recalculated, all the grid nodes in the smoothed grid are completely recalculated. This option can either increase or decrease the number of rows and columns in the smoothed grid relative to the original grid. The original grid values are lost unless their locations correspond exactly with the grid nodes in the output grid. The smoothed grid file is still an accurate representation of the original data.

Spline smoothing can be used to reduce grid file density. If a dense grid is created, producing a map from this grid might take a considerable amount of time. The less dense grid can be used to produce the contour map or wireframe in less time. USGS GTopo30 files are quite dense and it is necessary to thin them out before attempting to plot a map of the grid.

One of the purposes of spline smoothing is to fill in a sparse grid. A map produced from a sparse grid may have an angular appearance. For example, spline smoothing can be used to increase a 10 x 10 grid (a sparse grid) to a 50 x 50 grid. Denser grids produce smoother maps.



A 10 x 10 Grid



A 50 x 50 Grid

Increasing the grid density using spline smoothing increases the roundness or smoothness of the contours and the surface appearance.


Expanding a Grid


When a grid is expanded, the original grid nodes are preserved in the smoothed grid, and new grid nodes are added between existing grid nodes. The number of added grid nodes is defined by the *Number Nodes to Insert* boxes in the **Spline Smooth** dialog.

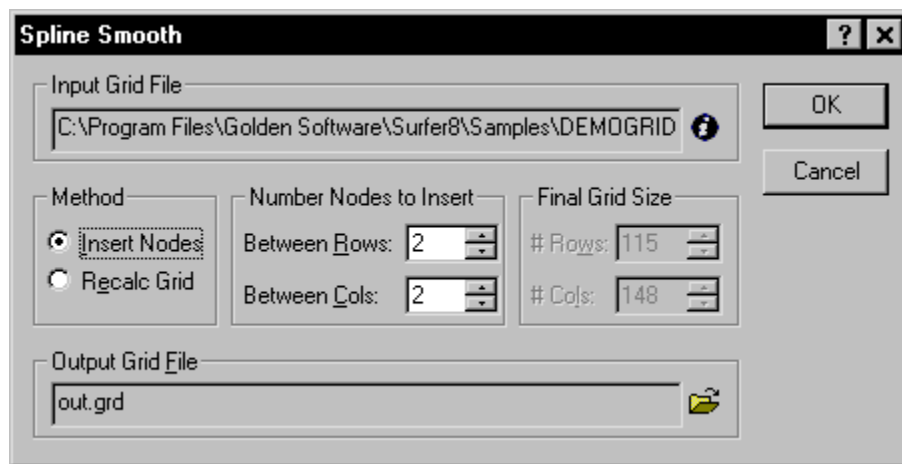
o	o	o	o	o	o	x	x	o	x	x	o	x	x	o	x	x	o	
						x	x	x	x	x	x	x	x	x	x	x	x	
						x	x	x	x	x	x	x	x	x	x	x	x	
o	o	o	o	o	o	o	x	x	o	x	x	o	x	x	o	x	x	o
						x	x	x	x	x	x	x	x	x	x	x	x	
						x	x	x	x	x	x	x	x	x	x	x	x	
o	o	o	o	o	o	o	x	x	o	x	x	o	x	x	o	x	x	o

A simple three by five grid is shown on the left in the above example. Each grid node is represented by an "o." When expanding the grid, two grid nodes are inserted between each existing grid node to produce the grid on the right. The original grid nodes are indicated with "o," and the new grid nodes are indicated with "x." The original grid is three by five and the smoothed grid is seven by 13.

To expand a grid:

1. Choose the **Grid | Spline Smooth** command.
2. In the **Open Grid** dialog, select the grid file to be smoothed.
3. Click *Open* to display the **Spline Smooth** dialog.
4. Choose the *Insert Nodes* option in the *Method* group to activate the *Number Nodes to Insert* group.
 - The *Between Rows* box specifies the number of rows to insert between the existing rows in the grid file. Rows correspond to grid nodes of constant Y.
 - The *Between Cols* box specifies the number of columns to insert between the existing columns in the grid file. Columns correspond to grid nodes of constant X.
5. Enter the number of nodes to insert, or use the up and down arrows to change the values. As you make the changes, the number of rows and columns to be produced in the smoothed grid is indicated in the *Final Grid Size* group.
6. If you need information on the original grid file, click the  button to display the number of rows and columns, minimum and maximum XYZ values, and statistics.

7. To change the name for the smoothed grid file, click the  button in the *Output Grid File* group, specify the path and file name in the **Save Grid As** dialog, and then click *Save*.
8. Click OK and the smoothed grid is created.





*To expand a grid, choose the Insert Nodes option in the **Spline Smooth** dialog.*

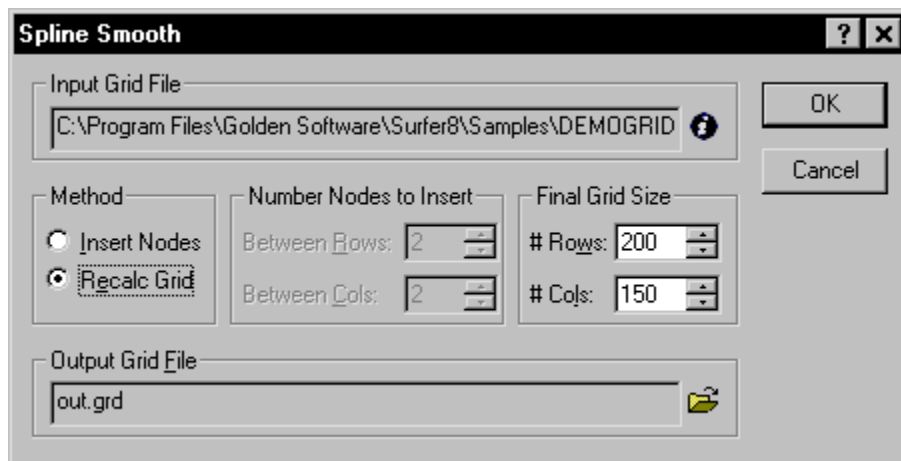
Recalculating a Grid

When a grid file is recalculated, the number of rows and columns are increased or decreased relative to the original grid. For example, you might have a 75 x 75 grid and need a 100 x 100 grid to perform grid math with another grid. You can recalculate the grid to have exactly the number of rows and columns needed.

To recalculate a grid:

1. Choose the **Grid | Spline Smooth** command.
2. In the **Open Grid** dialog, specify the grid file to be smoothed.
3. Click *Open* to display the **Spline Smooth** dialog.

4. Choose the *Recalc Grid* option in the *Method* group to activate the *Final Grid Size* group. This group specifies the number of rows and columns to produce in the smoothed grid. You can increase or decrease the number of rows and columns in the grid file.
 - The *# Rows* box specifies the number of rows for the smoothed grid file.
 - The *# Cols* box specifies the number of columns for the smoothed grid file.
5. Enter the desired value, or use the up and down arrows to change the values for the number of rows and columns to be produced in the smoothed grid.
6. If you need information on the original grid file, click the  button to display the number of rows and columns, minimum and maximum XYZ values, and statistics.
7. To change the name for the smoothed grid file, click the  button in the *Output Grid File* group, specify the path and file name in the **Save Grid As** dialog, and then click *Save*.
8. Click OK and the smoothed grid is created.



When recalculating a grid, choose the *Recalc Grid* option in the *Spline Smooth* dialog.

Grid Filtering

Grid filtering applies methods of digital image analysis to grids. This includes a broad suite of smoothing (low-pass) filters, as well as contrast enhancement filters, edge enhancement filters, edge detection filters, general high-pass filters, etc. **Surfer** also includes the capability for user-defined, general linear filters.

The Neighborhood

When filtering a grid, each node of the output grid is computed as a function of the corresponding node, and its neighbors, in the input grid. The concept of the neighborhood is used in grid filtering. The neighborhood of an output grid node is a rectangular sub-array of nodes in the input grid that is centered on the corresponding input grid node. A neighborhood has a non-zero width, and a non-zero height. Since the neighborhood is centered on a node, the width and height must both be odd numbers. For example, if the width and the height of the neighborhood are both three, the neighborhood of the output grid node at (21, 35) is the following rectangular sub-array of input grid nodes:

(20, 36) (21, 36) (22, 36)
(20, 35) (21, 35) (22, 35)
(20, 34) (21, 34) (22, 34)

If the height of the neighborhood is represented by S and the width of the neighborhood is represented by T, then the number of nodes in the neighborhood equals $S \times T$. Furthermore, the nodes in the neighborhood can be enumerated as:

$$\left\{ Z(r+i, c+j) \text{ where } i = -\left\lfloor \frac{S}{2} \right\rfloor, \dots, \left\lfloor \frac{S}{2} \right\rfloor \text{ and } j = -\left\lfloor \frac{T}{2} \right\rfloor, \dots, \left\lfloor \frac{T}{2} \right\rfloor \right\}$$

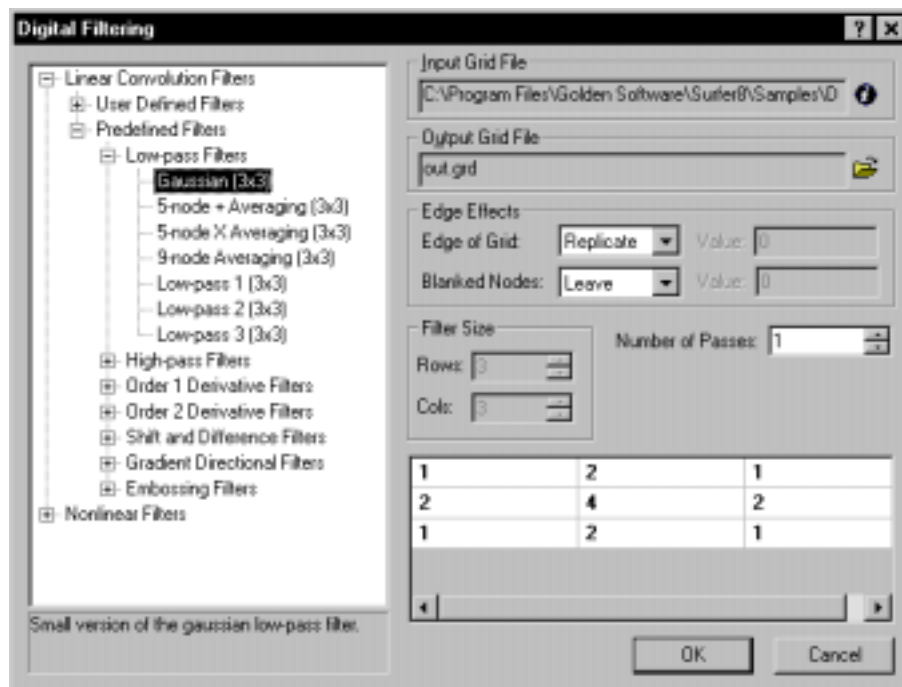
where $\lfloor A \rfloor$ is the largest integer less than or equal to A.

The box in the lower-right part of the dialog displays the neighborhood size, based on the number of *Rows* and *Cols*, along with the weights for each grid node in the neighborhood. Each element of the matrix is used to weight the grid node that lies "below" it. The products are then summed, normalized, and assigned to the value below the center node. The filter is then "moved" to the next node and the process is repeated until all nodes have been processed.

The Digital Filtering Dialog

Filter Categories


There are two main categories of filters: *Linear Convolution Filters* and *Nonlinear Filters*. Both of these general types are real space filtering methods. Refer to the *Linear Convolution Filters* and *Nonlinear Filters* sections for more information on these filters.



Select from a number of predefined filters or create your own filter in the **Digital Filtering** dialog.


Input Grid File

The *Input Grid File* lists the current grid file used in filtering. The grid must contain at least one non-blanked node.

The  button displays information about the grid file used to produce the filtered grid. The information includes the grid size, the minimum and maximum XYZ values contained in the grid file, and statistics. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.

Chapter 18 - Grid Operations

Digital filtering computes each node of the output grid as a function of the corresponding node, and its neighbors, in the input grid. Choose a path and file name for the grid in the *Output Grid*

File group. You can type a path and file name or click the  button to browse to a new path and enter a file name in the **Save Grid As** dialog.

Edge Effects

There are two settings for edge effects, that is, how to calculate the edge of the grid and how to calculate neighborhoods with blanked nodes. See *Edge of Grid* and *Blanked Nodes* for more information.

Filter Size

Filter Size is available with user-defined linear convolution filters and some nonlinear filters. You can use *Filter Size* to determine the size of the neighborhood.

Number of Passes

Set the number of times the filter is applied with the *Number of Passes* box.

Linear Convolution Filters

All linear convolution filters compute weighted averages of the neighboring input grid nodes. The only differences between the various linear convolution filters are the size and shape of the neighborhood, and the specific weights used. Consider the computation of the output grid value at row r and column c using a linear convolution filter with a neighborhood height S and width T .

$$\text{Numerator} = \sum_{i=-\left\lfloor \frac{S}{2} \right\rfloor}^{\left\lfloor \frac{S}{2} \right\rfloor} \sum_{j=-\left\lfloor \frac{T}{2} \right\rfloor}^{\left\lfloor \frac{T}{2} \right\rfloor} \begin{cases} 0 & \text{if } Z(r+i, c+j) \text{ is blank} \\ W(i, j)Z(r+i, c+j) & \text{otherwise} \end{cases}$$

$$\text{Denominator} = \sum_{i=-\left\lfloor \frac{S}{2} \right\rfloor}^{\left\lfloor \frac{S}{2} \right\rfloor} \sum_{j=-\left\lfloor \frac{T}{2} \right\rfloor}^{\left\lfloor \frac{T}{2} \right\rfloor} \begin{cases} 0 & \text{if } Z(r+i, c+j) \text{ is blank} \\ W(i, j) & \text{otherwise} \end{cases}$$

where $W(i, j)$ are the weights defined for the specified filter. The output grid node value is then

$$Z_{out}(r, c) = \begin{cases} \frac{\text{Numerator}}{\text{Denominator}} & \text{if } |\text{Denominator}| > 0 \\ \text{Numerator} & \text{otherwise} \end{cases}$$

User-Defined Filters

There are two types of user-defined filters, *Low-pass Filters* and *General User-defined*. With these filters, you can specify the height and width of the filter neighborhood.

Low-Pass Filters

A low-pass filter removes the high frequency noise with the resulting output being a smoother grid. There are four user-defined low-pass filters. Each of these four filters allows you to specify the size of the neighborhood. The width and height of the filter neighborhood must both be positive, odd numbers. Let the neighborhood height be S and width be T,

In the *Moving Average (mxn)* filter, the weights are all equal to one.

$$W(i, j) = 1 \quad \forall_i = -\left\lfloor \frac{S}{2} \right\rfloor, \dots, \left\lfloor \frac{S}{2} \right\rfloor \quad \forall_j = -\left\lfloor \frac{T}{2} \right\rfloor, \dots, \left\lfloor \frac{T}{2} \right\rfloor$$

With the *Distance Weighting (mxn)* filter, the weights fall-off with increased distance. The distance weighing function is

$$W(i, j) = \left(1 - \max\left(\frac{2|i|}{S+1}, \frac{2|j|}{T+1} \right) \right)^p \quad \forall_i = -\left\lfloor \frac{S}{2} \right\rfloor, \dots, \left\lfloor \frac{S}{2} \right\rfloor \quad \forall_j = -\left\lfloor \frac{T}{2} \right\rfloor, \dots, \left\lfloor \frac{T}{2} \right\rfloor$$

where p is the specified *Power*. The higher the power the more rapidly the weights fall-off with distance. The resulting iso-weight contour lines are concentric rectangles.

Chapter 18 - Grid Operations

With the *Inverse Distance (mxn)* filter, the weights fall-off with increased distance. With a neighborhood height S and width T , the distance weighing function is

$$W(i, j) = \begin{cases} W_c & \text{if } i = 0 \text{ and } j = 0 \\ \left(\frac{1}{i^2 + j^2} \right)^{p/2} & \text{otherwise} \end{cases} \quad \forall_i = -\left\lfloor \frac{S}{2} \right\rfloor, \dots, \left\lfloor \frac{S}{2} \right\rfloor \quad \forall_j = -\left\lfloor \frac{T}{2} \right\rfloor, \dots, \left\lfloor \frac{T}{2} \right\rfloor$$

where W_c is the specified *Central Weight* and p is the *Power*. The higher the power p , the more rapidly the weights fall-off with distance. The resulting iso-weight contour lines are concentric circles.

With the *Gaussian Low-pass (mxn)* filter, the weights fall-off with increased distance. With a neighborhood height S and width T , the distance weighing function is

$$W(i, j) = \exp \left(-\alpha \left(\left(\frac{i}{S} \right)^2 + \left(\frac{j}{T} \right)^2 \right) \right) \quad \forall_i = -\left\lfloor \frac{S}{2} \right\rfloor, \dots, \left\lfloor \frac{S}{2} \right\rfloor \quad \forall_j = -\left\lfloor \frac{T}{2} \right\rfloor, \dots, \left\lfloor \frac{T}{2} \right\rfloor$$

where α is the *Alpha* value (positive). This weight function takes the form of half the common bell-shaped curve. The parameter controls how quickly the weights fall-off with distance. The resulting iso-weight contour lines are concentric ellipses.

General User-Defined Filter

The *General User-defined (mxn)* linear filter allows you to specify the height and width of the filter neighborhood and any combination of weights. The box in the lower-right part of the dialog displays the neighborhood size, based on the number of *Rows* and *Cols*, along with the weights for each grid node in the neighborhood. Click in a cell in the box to change the node's weight.

The grid matrix can be selected and copied using CTRL+C and pasted into the user-defined matrix. This allows a pre-defined matrix to be used as a base and then be modified.

Predefined Filters

The predefined filters are a large collection of 3×3 filters defined in the references.

- *Low-pass Filters* are also known as smoothing or blurring filters. These filters remove the high frequency variation.
- *High-pass Filters* are also known as sharpening or crispening filters. They have the opposite effect of blurring. They tend to remove the background variation and emphasize the local details.
- *Order 1 Derivative Filters* are used to find horizontal and vertical edges.
- *Order 2 Derivative Filters* are another set of edge enhancement filters.
- *Shift and Difference Filters* are the two simplest horizontal and vertical differential operators.
- *Gradient Directional Filters* compute and return the directional derivatives in each of the eight compass directions.
- *Embossing Filters* identify and enhance edges aligned in one of the eight compass directions.

Nonlinear Filters

The *Nonlinear Filters* are not weighted averages of the neighboring input grid values; however, they are simple functions of the neighboring input grid values.

Order Statistics Filters

Filter	Description
<i>Minimum (mxn)</i>	The output grid node value equals the minimum of the neighboring values.
<i>Lower Quartile (mxn)</i>	The output grid node value equals the twenty-five percentile of the neighboring values.
<i>Median (mxn)</i>	The output grid node value equals the fifty percentile of the neighboring values.
<i>Upper Quartile (mxn)</i>	The output grid node value equals the seventy-five percentile of the neighboring values.
<i>Maximum (mxn)</i>	The output grid node value equals the maximum of the neighboring values.
<i>Range (mxn)</i>	The output grid node value equals the maximum minus the minimum of the neighboring values.

Moment Statistics Filters

Filter	Description
<i>Standard Deviation (mxn)</i>	The output grid node value equals the standard deviation of the neighboring values. The standard deviation is the square root of the variance (see below).
<i>Variance (mxn)</i>	<p>The output grid node value equals the variance of the neighboring values. The computation involves three steps. First, count the number of non-blank input grid node values in the neighborhood. With a neighborhood height S and width T, compute</p> $N = \sum_{i=-\lfloor S/2 \rfloor}^{\lfloor S/2 \rfloor} \sum_{j=-\lfloor T/2 \rfloor}^{\lfloor T/2 \rfloor} \begin{cases} 0 & \text{if } Z(r+i, c+j) \text{ is blank} \\ 1 & \text{otherwise} \end{cases}$ <p>Second calculate the local average, \bar{Z},</p> $\bar{Z} = \frac{1}{N} \sum_{i=-\lfloor S/2 \rfloor}^{\lfloor S/2 \rfloor} \sum_{j=-\lfloor T/2 \rfloor}^{\lfloor T/2 \rfloor} \begin{cases} 0 & \text{if } Z(r+i, c+j) \text{ is blank} \\ Z(r+i, c+j) & \text{otherwise} \end{cases}$ <p>Third, compute the local variance as</p> $\text{Variance} = \frac{1}{N} \sum_{i=-\lfloor S/2 \rfloor}^{\lfloor S/2 \rfloor} \sum_{j=-\lfloor T/2 \rfloor}^{\lfloor T/2 \rfloor} \begin{cases} 0 & \text{if } Z(r+i, c+j) \text{ is blank} \\ (Z(r+i, c+j) - \bar{Z})^2 & \text{otherwise} \end{cases}$
<i>Coef. of Variation (mxn)</i>	<p>The output grid node value equals the coefficient of variation of the neighboring values. The coefficient of variation is the standard deviation divided by the average. If the average (the denominator) is too small, then the coefficient of variation is arbitrarily set equal to 0.</p> <p>Coefficient of Variation = $\frac{\text{Standard Deviation}}{\bar{Z}}$</p>

Other Nonlinear Filters

Filter	Description
<i>Median Difference (mxn)</i>	<p>For each output grid node, (r, c), identify the set of non-blank, neighboring input grid node values. Compute the median of these neighboring values. Let B represent this median value. Then the output grid node value is set equal to</p> $Z_{\text{out}}(r, c) = Z(r, c) - B$ <p>The effect of this filter is to emphasize outliers in the grid.</p>
<i>Threshold Averaging (mxn)</i>	<p>For each output grid node, (r, c), identify the set of non-blank, neighboring input grid node values. Compute the average of these neighboring values, excluding the value of the corresponding input grid node. Let A represent this average value. Then the output grid node value is set equal to</p> $Z_{\text{out}}(r, c) = \begin{cases} Z(r, c) & \text{if } Z(r, c) - A \leq \text{Threshold} \\ A & \text{if } Z(r, c) - A > \text{Threshold} \end{cases}$ <p>where <i>Threshold</i> is the specified value. The effect of this filter is to eliminate outliers in the grid.</p>

Compass Gradient Filters

Compass gradient filters find edges in the eight different compass orientations (east, northeast, north, northwest, west, southwest, south, and southeast). First, eight different linear convolution filters designed to detect edges in each of the eight directions is applied. The output grid node value is the maximum absolute value resulting from these eight convolutions. The *Prewitt (3X3)*, *Kirsch (3X3)*, *Robinson 3-level (3X3)*, and *Robinson 5-level (3X3)* are four of the more common edge detection convolution filters in the literature. For details, see Crane (1997, p.88-90).

Edge of Grid

When computing the **Grid** | **Filter** values of the output nodes near the edge of the grid, the specified neighborhood may extend outside the range of the grid. This presents an issue because the output grid nodes are then functions of non-existent input grid nodes. There are three general ways of dealing with this issue: blank the output grid nodes near the edge, modify the defining function near the edge, or generate artificial input grid nodes beyond the edge upon which the defining function operates. The following methods for handling the edge are:

- *Blank* - Any grid node, for which its neighborhood overlaps one or more edge, is blanked. For example, if the neighborhood size is 3×3 then every application of the filter blanks one line of nodes on each edge. Every application of the filter shrinks the active grid by two rows and two columns: one row on the top, one row on the bottom, one column on the left edge, and one column on the right edge. Similarly, a 5×5 filter neighborhood blanks four rows and four columns every application.
- *Ignore* - This is the default setting. In this case, the filter function is modified by truncating the neighborhood at the edge of the grid. For example, in *The Neighborhood* section, even though the general neighborhood is specified to be 3×3, the effective neighborhood of node (21, 0), on the left edge of the grid, is only 3×2:

$$\begin{array}{l} (22, 0) (22, 1) \\ (21, 0) (21, 1) \\ (20, 0) (20, 1) \end{array}$$

This approach to handling the edges applies an essentially different function near the edges than in the middle of the grid. This can cause unexpected, visually apparent, artifacts in some extreme cases.

- *Replicate* - The edge grid node value is copied. For a grid with M rows and N columns, the rules are

$$Z(r, c) = Z(a, b)$$

where

$$a = \begin{cases} 0 & \text{if } r < 0 \\ r & \text{if } 0 \leq r < M \\ M - 1 & \text{if } r \geq M \end{cases} \quad b = \begin{cases} 0 & \text{if } c < 0 \\ c & \text{if } 0 \leq c < N \\ N - 1 & \text{if } c \geq N \end{cases}$$

- *Mirror* - The grid node value pattern at the edge is mirrored. For a grid with M rows and N columns, the rules are

$$Z(r, c) = Z(a, b)$$

where

$$a = \begin{cases} -r & \text{if } r < 0 \\ r & \text{if } 0 \leq r < M \\ 2M - 2 - r & \text{if } r \geq M \end{cases} \quad b = \begin{cases} -c & \text{if } c < 0 \\ c & \text{if } 0 \leq c < N \\ 2N - 2 - c & \text{if } c \geq N \end{cases}$$

- *Cyclic Wrap* - The grid is wrapped in two dimensions. If you go off the grid on the right edge, you come back on the left; if you go off on the top, you come back on the bottom, etc. For a grid with M rows and N columns, the rules are

$$Z(r, c) = Z(a, b)$$

where

$$a = \begin{cases} r + M & \text{if } r < 0 \\ r & \text{if } 0 \leq r < M \\ r - M & \text{if } r \geq M \end{cases} \quad b = \begin{cases} c + N & \text{if } c < 0 \\ c & \text{if } 0 \leq c < N \\ c - N & \text{if } c \geq N \end{cases}$$

- *Fill* - Fill the edge with a specified constant. A common value for the fill is the arithmetic average of the grid.

$$Z(r, c) = \begin{cases} Z(r, r) & \text{if } 0 \leq r < M \text{ and } 0 \leq c < N \\ A & \text{otherwise} \end{cases}$$

where A is the user-specified constant. Enter a number into the *Value* box to the right when using this option.

Blanked Nodes

There are several ways to handle blanked nodes in a neighborhood in filtering. (If all nodes in a neighborhood are blanked, the output node is also blanked.)

- *Expand* the blanked regions. If the neighborhood of an output grid node contains a blanked node in the input grid, then the output node is blanked. Like blanking on the edge, this approach leads to blanked areas that grow with every application of the filter.
- *Leave* the blanks. This is the default setting. Blank every output grid node for which the corresponding input grid node is blank. When the corresponding input grid node is not blank, but a neighboring grid node is blank, modify the filter to ignore the blank; essentially, remove the blank node from the neighborhood. For example, if the filter called for computing the median value in the neighborhood, the blanked values would not be considered when determining the median. This keeps the blanked areas constant, but can cause internal artifacts for some filter types.
- *Ignore* the blanks by filtering across them. The blank nodes are removed from the neighborhood. For example, if the filter called for computing the median value in the neighborhood, the blanked values would not be considered when determining the median. This option is similar to *Leave*, however, this option does not blank the output grid nodes corresponding to blank input grid nodes. This is essentially a simultaneous filtering and interpolation. Every application of the filter would see a shrinking of the blanked regions, since the only blank output grid nodes are those with completely blank neighborhoods.
- *Fill* blanks with a user-specified constant prior to filtering. When using this option, enter the *Fill* number into the *Value* field.

Grid Filter References

Crane, R. (1997) *A Simplified Approach to Image Processing: Classical and Modern Techniques in C*, Prentice Hall PTR, Upper Saddle River, NJ, 317 pp. ISBN: 0-13-226416-1.

Pitas, I. (2000) *Digital Image Processing Algorithms and Applications*, John Wiley and Sons, New York, 419 pp. ISBN: 0-471-37739-2.

Wiggin, Ender (2001) *Elementary Digital Filtering*,
<http://www.gamedev.net/reference/articles/article1068.asp>, (09/20/2001) .

Grid Mosaic

The **Grid | Mosaic** command combines two or more input grids of the same coordinate system into a single output grid. For example, if you have four USGS SDTS Raster Profiles (DEMs), you can easily combine them into one grid with **Grid | Mosaic**.

Grid Mosaic Dialog

Input Grid Files

The *Input Grid Files* group initially contains all the grid files selected in the **Open Grid** dialog. Use the *Add* button to add more grids to the mosaic and use the *Remove* button to remove grids from the mosaic. All grids must be in the same coordinate system. This means that 30-meter DEM files must be in the same UTM zone.


The header buttons can be dragged side to side to make the columns wider or narrower. In addition, the grids can be sorted by clicking on the *Grid Filename*, *xMin*, *xMax*, *yMin*, and *yMax* headers. Alternatively, the grids can be moved around the list by highlighting a grid name (click on it) and then clicking the *Up* and *Down* buttons.

Click the *Info* button to obtain information about the selected grid, including the grid file name and statistics. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.

There are three *Resample Methods* to choose from: *Bilinear Interpolation*, *Nearest Neighbor*, and *Cubic Convolution*.

When grids overlap, choose an *Overlap Method* to determine the value of the grid node in the new grid: *Average*, *First*, *Last*, *Minimum*, or *Maximum*.

Output Grid File

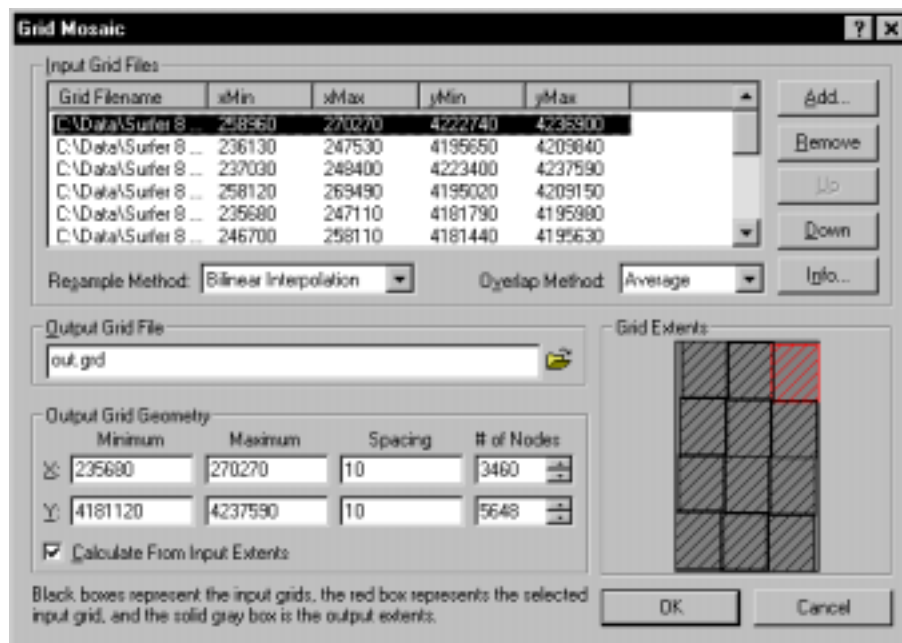
Choose a path and file name for the grid in the *Output Grid File* group. You can type a path and file name or click the  button to browse to a new path and enter a file name in the **Save Grid As** dialog.

Chapter 18 - Grid Operations

Grid Line Geometry

Output Grid Geometry defines the grid limits and grid density. Grid limits are the minimum and maximum X and Y coordinates of all grid files. The default grid *Spacing* is set to the minimum spacing of the input grids. The *# of Nodes* in the X direction is the number of grid columns, and the *# of Nodes* in the Y direction is the number of grid rows. Increasing the grid extents does not increase the grid as the resampling methods do not extrapolate beyond the extents of the original grid.

Check the *Calculate From Input Extents* box to return the grid to its original size based on the input grids' minimum and maximum.



Use *Grid / Mosaic* to join adjacent grids using the same coordinate systems.

Grid Extents

The graphic represents input grids with black boxes. The selected input grid is in red, and the output extents are represented by the solid gray box. The input grid boxes can be clicked with the mouse to graphically select the grid.

Resample Methods

When using **Grid | Mosaic, Surfer** uses resampling methods to assign data values in the new grid. For each grid node location in the output grid (X,Y) the Z value is located in the set of input grids. The Z value is computed according to the specified resample method.

Note that all resample methods are exact interpolators - that is, if the X,Y location falls exactly on a grid node, the grid node's Z value is returned. If more than one grid can supply a Z value at the specified X,Y position, they are combined according to the specified *Overlap Method*.

The resampling methods do not extrapolate outside the grid limits. This means that blanked nodes may be inserted between adjacent input grids if the input grids do not each include the common boundary.

Nearest Neighbor

The nearest neighbor method applies the closest grid node value on the original grid to the grid node value in the new grid. When the original grid and the new grid differ in size, more than one original node may be applied to the new grid and some original grid cells may not be applied to the new grid.

The nearest neighbor method is the fastest resampling method, though it can result in distorted output if the original grid and new grid differ in size.

Bilinear Interpolation


The bilinear interpolation method uses a weighted average of four nodes in the original grid and applies this to the new grid node. The new grid is smoothed compared to the original grid.

Cubic Convolution

The cubic convolution method uses a weighted average of 16 nodes in the original grid and applies this to the new grid. The new grid is smoother than the original grid. This method is best for continuous data. This is the slowest resampling method, but it results in a sharper grid than the bilinear interpolation or the nearest neighbor methods.

Joining Grids

To combine grids:

1. Click **Grid | Mosaic**.
2. Select the files you would like to combine in the **Open Grid** dialog. If the grid files are in the same folder, hold the CTRL and SHIFT keys down while clicking on the file names to make multiple selections. If the files are not in the same folder, you will have the opportunity to select more files later.
3. Click *Open* in the **Open Grid** dialog to open the **Grid Mosaic** dialog.
4. Choose mosaic options in the **Grid Mosaic** dialog.
5. Name the new grid in the *Output Grid File* box by typing a path and file name or using the  button to browse to a new path.
6. Click OK and the new grid is created.

Mosaic Tips

- When a blanked node overlaps a non-blanked node, the blanked node is ignored.
- The default output grid spacing is set to the minimum spacing of the input grids.
- By specifying *First* for the *Overlap Method*, a second overlapping grid can be used to fill in blanked values in the first grid.
- Some grid files, such as USGS DEM files, are not seamless. Occasionally, a grid node is missing along an edge. Sometimes, the reported corner points for adjacent files are slightly different, which causes the neat lines between grids to be slightly off. This causes some edge nodes to be excluded from both grids. These areas show up as a blanked node along the seam.
- It is possible to resample a single grid with **Grid | Mosaic** by specifying a single grid and changing the grid spacing.

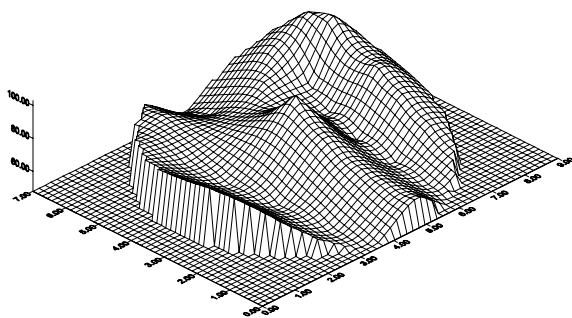
Blanking a Grid File

The **Grid | Blank** command is used to remove grid node data from a grid in areas not supported by original data or in areas where you do not want to display contours on a map. Blanking assigns a blanking value ($1.70141\text{e}+038$) to specified groups of grid nodes in a blanked grid file. Blanked grids contain the same number of grid nodes and the same grid limits as the original grid file. The grid nodes in the blanked grid are identical to the values in the original grid except in those locations where the blanking values are assigned.

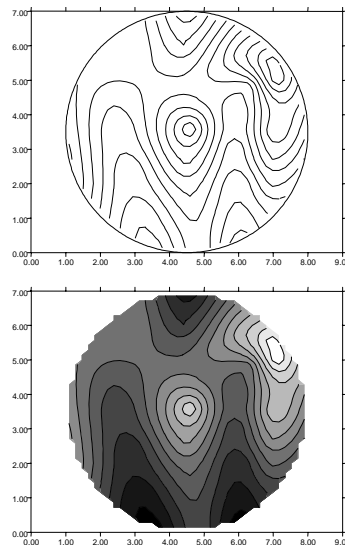
Blanking boundaries are defined in a blanking file, which is a special ASCII format file containing the XY coordinates defining the boundary and header information. Refer to *Appendix C, File Formats* for the blanking file format. Blanking can be assigned to areas inside or outside the blanking boundary. Alternatively, blanking regions can be specified using the **Options | Blank Node** command in the **Grid Node Editor** window. See the *Grid Node Editor* section of this chapter for more information.

The process of blanking marks areas of a grid or map as "no data" areas. In blanked areas of a grid:

- Contours are not drawn.
- A separate fill color can be assigned to blanked areas in contour maps, image maps, and shaded relief maps.
- Wireframes display a flat surface at the minimum Z value in the grid file.
- Volumes are not calculated and areas are calculated separately.
- **Grid | Math** operations produce a blanked node if one of the grids contains a blanked node at a given location.



Blanked regions appear as low flat regions on a wireframe. This wireframe is produced from the same grid file used to produce the contour maps on the right.



The upper map shows unfilled contours and a circular blanking boundary. Contour lines are truncated within one grid cell width of the blanking boundary. The lower map shows a color-filled contour map with a white fill assigned to blanked areas.

Chapter 18 - Grid Operations

There are several ways that blanked areas are created within a grid.

Starting with a data file, the **Grid | Data** command blanks grid nodes:

- where the search criteria are not satisfied.
- outside the convex boundary of the data set if the triangulation or natural neighbor gridding methods are used.
- within an area delineated by a fault polygon if there are no data values within that area.

Starting with a grid file, nodes are blanked if:

- the **Grid | Blank** command is used to blank an area inside or outside a boundary defined by a blanking file.
- the **Grid | Grid Node Editor** is used to blank individual grid nodes.

Blanking Inside or Outside a Polygon


The **Grid | Blank** command blanks areas of a map inside or outside a polygonal boundary. It creates a new grid file with blanking values assigned to grid nodes without changing the number of grid nodes or the XY grid limits from the original grid file.

The boundary of the blanked area is defined by the blanking file, an ASCII format defined in *Appendix C, File Formats*. Blanking files can be created in the **Surfer** worksheet, in a text editor, or by using **Map | Digitize**.

To blank a grid file:

1. Create a blanking file. Blanking files can be created in the worksheet or text editor (see *Appendix C, File Formats*) or they can be created with **Map | Digitize** (see *Chapter 7, Base Maps*).
2. Choose the **Grid | Blank** command.
3. Choose the input grid file in the **Open Grid** dialog, and then click *Open*.
4. Choose the blanking file in the **Open** dialog, and then click *Open*.
5. Name the output grid file in the **Save Grid As** dialog, using a different name than the input grid file, and then click *Save*.

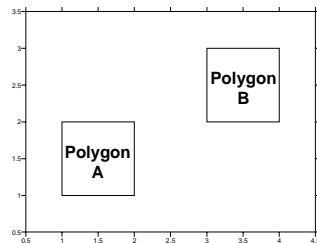
The new grid file is created without changing the plot window display. To create a new map with the new grid file, choose a command from the **Map** menu. To substitute the new grid file in an

existing map, double-click on the map, click on the  button, and choose the new grid file. To display the blanking file on the new map, choose the **Map | Base Map** command, select the blanking file name in the **Import File** dialog, and then click *Open*.

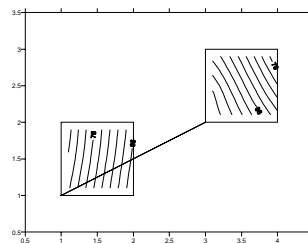
Blanking Outside Multiple Polygons

Blanking the area outside multiple polygons can present a problem. If the area outside polygon A includes the area inside polygon B, and the area outside polygon B includes the area inside polygon A, then the entire grid is blanked.

The problem can be resolved by combining the two polygons into a single polygon. Edit the blanking file in the **Surfer** worksheet or in a text editor, repeating the first polygon vertex at the end of each polygon, and editing the header to reflect the total number of vertices. The new blanking file is best suited for blanking, with the original blanking file used for display as a base map. See *Appendix C, File Formats* for more information about the blanking file format.



Blanking the area outside multiple polygons blanks the entire grid.



Editing the blanking file to combine the polygons resolves the problem.

```
5 0      "Pol ygon  A"
1.0 1.0
1.0 2.0
2.0 2.0
2.0 1.0
1.0 1.0
5 0      "Pol ygon  B"
3.0 2.0
4.0 2.0
4.0 3.0
3.0 3.0
3.0 2.0
```

```
11 0
1.0 1.0 "Pol ygon  A"
1.0 2.0
2.0 2.0
2.0 1.0
1.0 1.0
3.0 2.0 "Pol ygon  B"
4.0 2.0
4.0 3.0
3.0 3.0
3.0 2.0
1.0 1.0 "Repeat 1st
vertex"
```

*The blanking file on the left contains two separate polygons.
The blanking file on the right contains two combined polygons
that can be used to blank outside of two areas.*

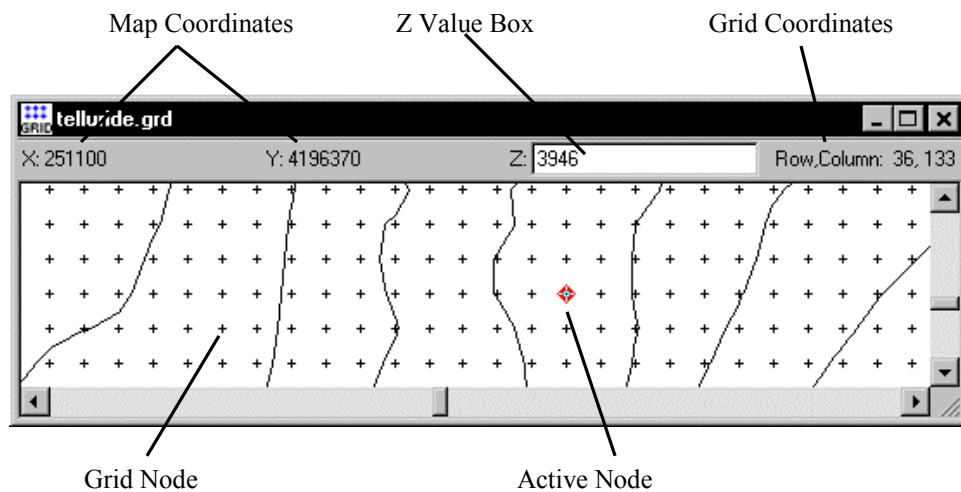
Blanking Individual Grid Nodes

The **Grid | Grid Node Editor** command controls the blanking of individual grid nodes. It is described in the following section, *Grid Node Editor*.

Grid Node Editor

The **Grid | Grid Node Editor** command allows you to change or blank Z values at individual grid nodes in a grid. Each grid node is indicated with a + in the grid editor window when the **Options | Show Nodes** menu option is selected. The active node is highlighted with a red diamond. To move between grid nodes, press the arrow keys, or click a node to make it the active node.

The active node XY map coordinates and grid coordinates are displayed at the top of the window, and the Z value is given in the Z box. You can enter a new Z value for the selected grid node into the box. Press ENTER, an arrow key, or click another node with the mouse to enter the new value into the grid. The contour map is redrawn with your change when the **Options | Show Contours** menu option is selected. You can save the edited grid file with the same name or a different name.



The grid node editor modifies individual grid nodes in a grid file. Grid nodes are indicated with the + symbol, and the active grid node is highlighted with a red diamond. The XY map coordinates, grid coordinates, and Z value box are displayed at the top of the window.

The Options Menu

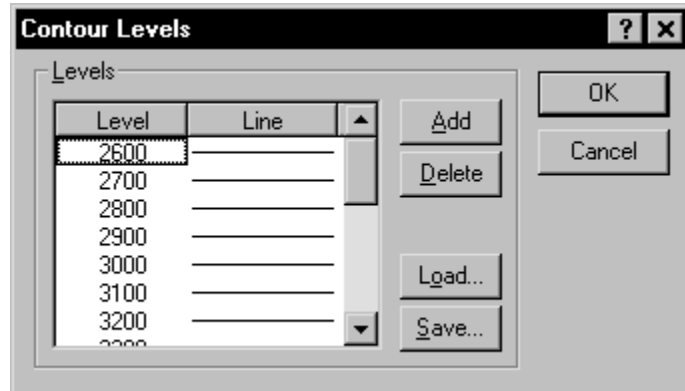
The grid editor **Options** menu contains commands that control the display of the grid nodes and the contour map in the grid node editor window.

The **Show Contours** command controls the display of contour lines on the map. When the command is checked, the contours are displayed on the map. Click on the command to toggle between showing and hiding contours. As individual grid nodes are modified, the contours are redrawn to reflect the change.

The **Contour Levels** command displays the **Contour Levels** dialog, allowing you to control the display of the contours.

The *Levels* group lists the contour levels and colors of contour lines used in the grid editor window.

- Click the *Level* button to display the **Contour Levels** dialog. The **Contour Levels** dialog controls the minimum and maximum contour levels as well as the contour interval to use between levels.
- The *Line* button displays the **Line Spectrum** dialog that controls the colors and widths of the minimum and maximum contour lines. **Surfer** automatically produces a gradational spectrum of colors between the minimum and maximum colors.
- The *Add* button adds a contour line halfway between selected contour level and the next lower contour level. If the minimum contour level is selected, the added contour level is the same as the minimum level.
- The *Delete* button removes the selected contour from the list of contours.
- The *Load* button allows you to select an existing level file containing contour level information.
- The *Save* button allows you to save the current contour level and line property information to a level file.



*The **Contour Levels** dialog controls the display of the contour map in the grid editor window.*

The **Show Nodes** command turns on the display of grid node markers on the map. When the command is checked, a + appears at the location of each grid node. When the command is not checked, the crosses are not shown, but you can still edit the grid. Some node markers are not displayed if the zoom or density is such that they are too close. Although not all of the markers are shown, the active node indicator still "snaps" to each grid node as the arrow keys and mouse are used.

The **Blank Node** command assigns the blanking value (1.70141e+038) to the selected node. Contour lines are not drawn in the blanked regions of grid files. When you display a blanked grid file as a wireframe, the blanked regions appear as low, flat regions at the same level as the minimum Z value in the blanked file.

Chapter 18 - Grid Operations


The **Unblank Node** command is grayed unless a grid node blanked during the current grid node editor session is selected. During the session, if a grid node is blanked, the original Z value for that node is preserved in memory, so you can return the grid node to its pre-blanked value. If the grid file you are editing was blanked before you opened the grid editor window, you cannot unblank the node with this command.

Click the **Grid Info** command to obtain information about the grid, including the grid file name and statistics. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.

Using the Grid Node Editor

To edit grid nodes:

1. Select **Grid | Grid Node Editor** in the plot window.
2. In the **Open Grid** dialog, select a grid file and then click the *Open* button.
3. The grid file is represented by a contour map, and the location of each grid node in the file is marked with a +.
4. The active node is highlighted with a red diamond. To move between grid nodes, press the arrow keys, or click a node to make it the active node.
5. The active node XY map coordinates and grid coordinates are displayed at the top of the window, and the Z value is given in the Z box. You can enter a new Z value for the selected grid node into the box. Press ENTER or click another node with the mouse to enter the new value into the grid. The contour map is redrawn with your change when the **Options | Show Contours** menu option is selected.
6. To blank a node, select **Options | Blank Node** or press CTRL+B.
7. To return to the original value, press the ESC key while the cursor is in the Z value box.
8. When you are done editing the grid, choose the **File | Save** command to save the file to the same name, or the **File | Save As** command to specify a new file name.
9. To close the grid editor window, choose the **File | Close** command. To move to an existing window and leave the grid editor window open, click the **Window** menu and select a different window.

You can use the new grid by selecting a command from the **Map** menu, or by clicking the  button for an existing map.

Grid Convert

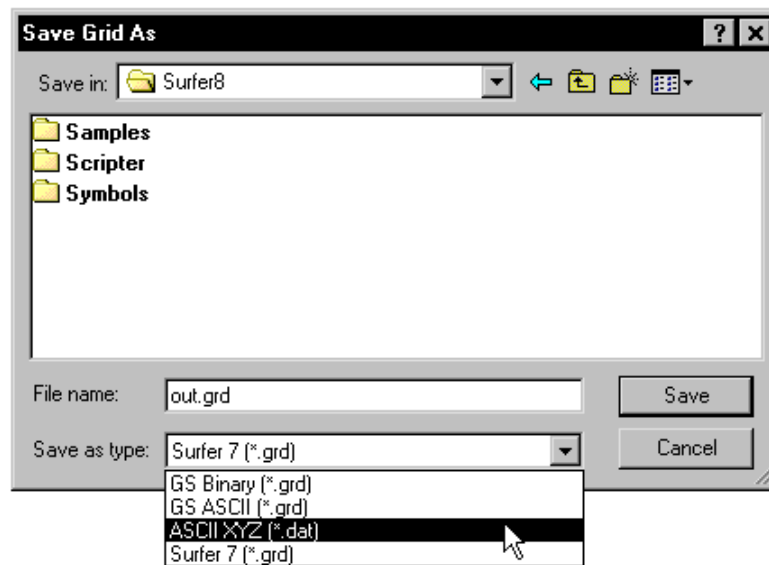
The **Grid | Convert** command converts grid files to another grid format, or an ASCII XYZ format. The ASCII XYZ format consists of separate rows with the X coordinate in Column A, the Y coordinate in Column B, and the corresponding Z value in Column C. **Surfer** does not export as any of the USGS or DTED file formats. **Surfer** supports three export grid file formats:

- The default *Surfer 7 (*.grd)* format is a binary format that uses double precision values and includes fault information used to create the grid. It cannot be read by older versions of **Surfer**.
- The *GS Binary (*.grd)* format is a binary format that uses single precision values and is compatible with current and older **Surfer** versions. This format produces the smallest grid file size. There is no fault information included in this format.
- The *GS ASCII (*.grd)* format is an ASCII version of the *GS Binary (*.grd)* format. It uses single precision values and is compatible with current and older versions of **Surfer**. This format produces a larger file size than the *GS Binary (*.grd)* format, but it can be edited in the **Surfer** worksheet or a text editor.

Converting Grid File Formats

To convert a grid (**Surfer** grid, USGS DEM, or a DTED):

1. Choose the **Grid | Convert** command to display the **Open Grid** dialog.
2. Select the grid file and click the *Open* button.
3. The **Save Grid As** dialog prompts for the new file name and format. In the *Save as type* list, select the format for the converted file. The options are *GS Binary (*.grd)* format, *GS ASCII (*.grd)* format, *ASCII XYZ (*.dat)*, and *Surfer 7 (*.grd)* format.
4. Specify the path and file name for the converted file and then click *Save* to create the new file.



You can select a file format type from the Save as type list.

Grid Extract

The **Grid | Extract** command creates a subset of an existing grid file. Grid files consist of rows and columns (lines of constant Y and X value) containing Z values. When you extract information from a grid file, you can specify the starting and ending rows and columns to extract. For example, you can create a new grid file that consists of only the center portion of an existing grid by specifying the middle columns and rows from the original grid.

Consider a grid file that has 100 rows and 100 columns. You might want to create a grid that contains the information from row 25 to row 75 and column 25 to column 75. In this example, specify 25 as the first row and first column to extract and 75 as the last row and last column to extract. The grid file you create in this case would consist of 51 rows and 51 columns of data (from 25 to 75 rows and columns, inclusive).

Subsets can also be based on every n^{th} row or column read from the input grid file, reducing the density of the grid. In this case, you specify a step factor that skips rows and columns when reading information from the original grid file.

Extract Grid

Input File
C:\Program Files\Golden Software\Surfer8\Samples\DEMOGRID.GRD

Direction	# of Nodes	Minimum	Maximum	Spacing
Rows (Y):	39	0	7	0.18421053
Cols (X):	50	0	9	0.18367347

Output File
out.grd

Direction	First	Last	Read Every	# of Nodes	Minimum	Maximum
Rows (Y):	1	39	1	39	0	7
Cols (X):	1	50	1	50	0	9

*The **Extract Grid** dialog displays information about the input grid file, lets you specify a subset of a grid, and writes the subset to a new grid file.*


The Extract Grid Dialog

In the **Extract Grid** dialog, the *First* and *Last* boxes show the first and last rows and columns from the input grid file. The number of nodes and the map coordinates are displayed to the right. As you change the *First* and *Last* values, the number of nodes and map coordinates change accordingly.

The *Read Every* boxes specify a skip factor to "thin out" a grid. A value of one reads every row or column within the selected limit. A value of two reads every other row or column, and so on.

Extracting Grids

To extract a subset of an existing grid file:

1. Choose the **Grid | Extract** command to display **Open Grid** dialog.
2. Select the grid file from which you want to extract a subset and click *Open* to display the **Extract Grid** dialog.
3. Specify the *First*, *Last*, and *Read Every* values in the *Rows (Y)* and *Cols (X)* boxes.
4. Click the  button to display the **Save Grid As** dialog.
5. In the *Save as type* drop-down list, select the format for the output file.
6. Specify the path and file name for the file to be created, and click *Save*. The grid or data file is created with the subset of grid information from the original grid file.


Grid Transform

The **Grid | Transform** command contains several options that modify the X and Y map coordinates of grid nodes within the grid file; it does not alter the Z values. The **Transform** options are used to shift, scale, rotate, or mirror grid nodes within a grid file.

The Transform Dialog

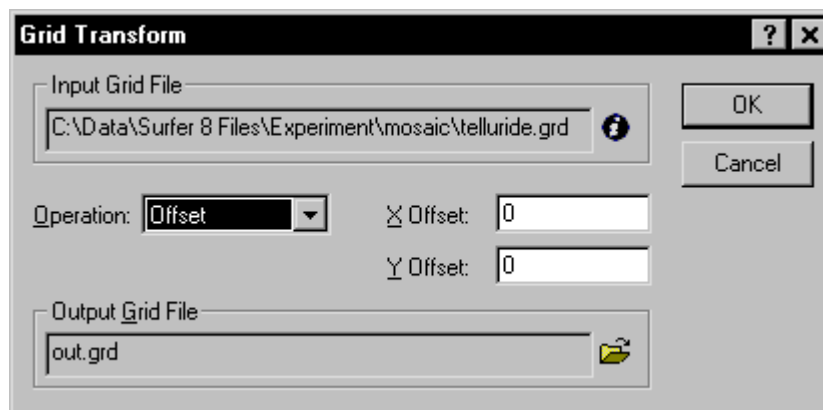
Input Grid File

The *Input Grid File* lists the grid file to be transformed. The grid must contain at least one non-blanked node.

The  button displays information about the grid file used in the transformation. The information includes the grid size, the minimum and maximum XYZ values contained in the grid file, and statistics. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.

Operation

The *Operation* list controls the transformation type. Click on the box or drop-down arrow and select an option from the list.



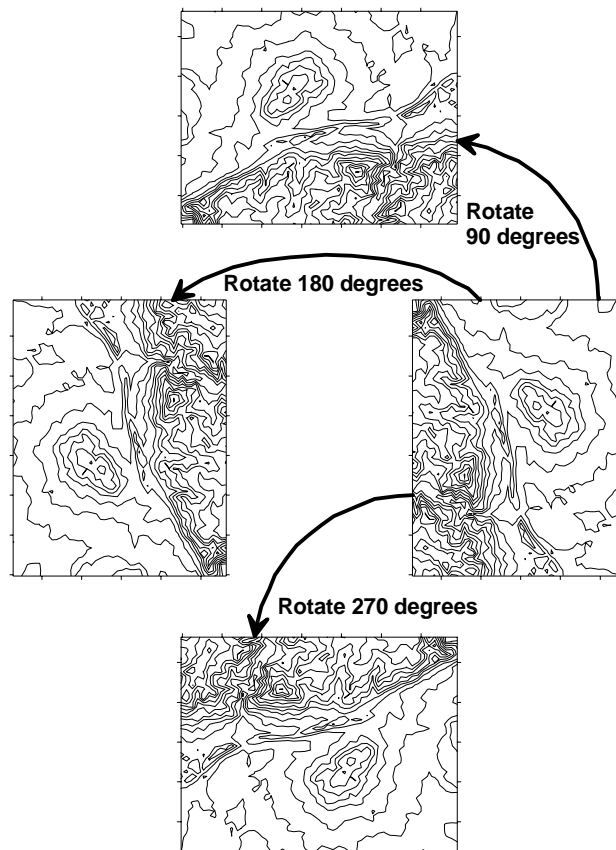
*The **Grid Transform** dialog controls the type of transform and the parameters for each transform.*

The *Offset* option adds or subtracts a number from the X and Y dimensions. Specify the *Offset* option from the *Operation* list, and then set the amount of the offset in the *X Offset* and *Y Offset* boxes in positive or negative units. For example, if the input grid ranges are X: 1 to 9 and Y: 2 to 8, and the offsets are 10.0 for both the *X Offset* and *Y Offset*, then the output grid ranges will be X: 11 to 19 and Y: 12 to 18.

The *Scale* option multiplies the X and Y ranges by a factor. Specify the *Scale* option from the *Operation* list, and then specify the scaling factors in the *X Scale* and *Y Scale* boxes displayed in the **Grid Transform** dialog. The scaling origin is the lower left corner of the grid, so only the *xMax* and *yMax* values are changed. For example, if the input grid ranges are X: 1 to 9 and Y: 2 to 8 and the *X Scale* and *Y Scale* factors are both 3.0, then the output grid ranges will be X: 1 to 25 $((9 - 1) \times 3.0) + 1$ and Y: 2 to 20 $((8 - 2) \times 3.0) + 2$.

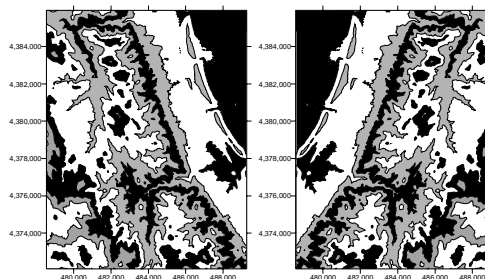
Chapter 18 - Grid Operations

The *Rotate* option rotates a grid in multiples of 90 degrees (0, 90, 180, 270, ...). Select the *Rotate* option in the *Operation* list, and specify the rotation angle in the *Angle* box. The origin of the map remains in the lower left, and the axes continue to increase in the same direction, with the X axis increasing to the right, and the Y axis increasing upwards.



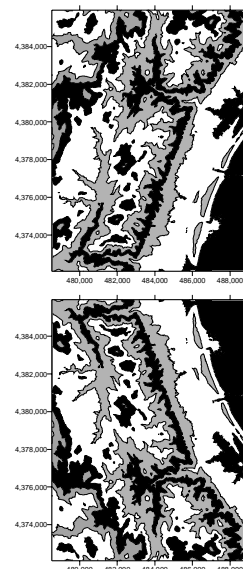
The Rotate option rotates the grid in multiples of 90 degrees without changing the XY origin.

The *Mirror X* option creates a new grid file with the order of the grid nodes reversed in the X direction. The output grid file uses the same minimum and maximum X and Y coordinates as the input grid file. The origin of the grid remains in the lower left corner, with X increasing to the right and Y increasing upward. The Y order remains unchanged.




The Mirror X option creates a mirror image of the grid file with the X order reversed.

The *Mirror Y* option creates a new grid file with the order of the grid nodes reversed in the Y direction. The output grid file uses the same minimum and maximum X and Y coordinates as the input grid file. The origin of the grid remains in the lower left corner, with X increasing to the right and Y increasing upward. The X order remains unchanged.




The Mirror Y option creates a mirror image of the grid file with the Y order reversed.

Output Grid File

In the *Output Grid File* group click the  button to specify a different path or file name for the grid file to be created.

Transforming a Grid File

To transform a grid file:

1. Choose the **Grid | Transform** command to display the **Open Grid** dialog.
2. Select a grid file to transform and click the *Open* button.
3. In the **Grid Transform** dialog, choose an *Operation* and any associated options.
4. Click the  button to display the **Save Grid As** dialog. In the *Save as type* list, select the format for the output file. The options are *GS Binary (*.grd)*, *GS ASCII (*.grd)*, *ASCII XYZ (*.dat)*, and *Surfer 7 (*.grd)* formats.
5. Enter the path and file name for the transformed grid.
6. Click *Save* to create the transformed file.

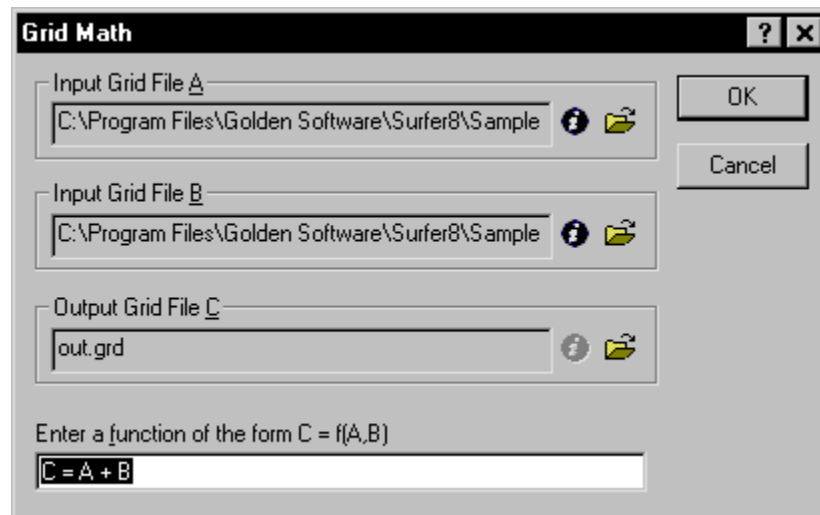
Grid Math

The **Grid | Math** command creates a new grid file that transforms the Z values of a single grid file or combines Z values from two grid files that use the same XY limits and grid dimensions. The output grid file is based on a mathematical function of the form $C = f(A,B)$, where C is the output grid file, and A and B represent input grid files. The defined function is performed on corresponding nodes with the same XY coordinates from the input grid files, and the result of the calculation is placed at the same coordinates in the output grid file. For example, the function $C = \log_{10}(A)$ creates an output grid file with the logarithmic base 10 values for each of the grid nodes in Grid A. If a node is blanked in either input grid, it is blanked in the output grid.

Using the Grid Math Command


To convert a grid using **Grid | Math**:

1. Choose the **Grid | Math** command to display the **Open Grid** dialog.
2. Select one of the input grid files, and click the *Open* button.





*The **Grid Math** dialog controls the input and output grid files and the function used to calculate the output grid file.*

3. In the **Grid Math** dialog, the *Input Grid File A* group displays the grid file selected in step 1.

The X and Y limits and grid size of this file define the output grid. Click the  button to display the grid rows, grid columns, XYZ minimums and maximums, and statistics. Click the





button to change the input grid file.

4. In the *Input Grid File B* group, click the  button to choose an optional second grid file for the grid math operation. Both grids must contain the same number of grid rows and columns, and should extend over the same X and Y ranges. To perform an operation on *Input Grid File A* only, you can leave this group set to *None*.
5. In the *Output Grid File C* group click the  button to specify a different path or file name for the grid file to be created.
6. In the *Enter a function of the form $C = f(A,B)$* box, enter a function where C represents the output grid file, and A and B refer to the input grid files. To perform a calculation on grid file A only, you can leave any reference to grid file B out of the specified function. (See *Appendix A, Mathematical Functions* for the available functions.)
7. Click OK to create the new grid file.

Using the Grid Math Command to Generate an Isopach Map

The **Grid | Math** command can be used to generate an isopach map. For example, let's say you have drill hole data that provides the elevation of the top and bottom of a coal seam, and want to generate a contour map representing the thickness of the seam. The elevation of the top of the coal seam can define the upper surface, and the elevation of the base of the coal seam could define the lower surface. Using grid math to subtract the lower surface from the upper surface would generate a grid file that represents the thickness of the coal seam.


To create an isopach map:

1. Create grid files of the upper and lower surfaces. In the example here, the grid files would define the elevation of the top and bottom of the coal seam. Both grids must use the same XY ranges, and same number of rows and columns.
2. Choose the **Grid | Math** command.
3. Select the grid file for the upper surface in the **Open Grid** dialog.
4. In the **Grid Math** dialog, the grid file selected in step 3 is shown in the *Input Grid File A* group.
5. In the *Input Grid File B* group, click the  button, select the grid file for the bottom surface in the **Open Grid** dialog, and then click *Open*.
6. Click the  button in the *Output Grid File C* group, enter the name for the thickness grid file in the **Save Grid** dialog, and then click *Save*.
7. In the *Enter function of a form $C = f(A,B)$* box, enter the formula $C = A - B$. Click OK and a grid file is created which can be used to generate the isopach map of the thickness between the two input grid files.
8. To view a contour map of the thickness grid file, choose the **Map | Contour Map | New Contour Map** command and choose the grid file you created with the **Grid Math** command.

Using Grid Math to Eliminate Negative Values from a Grid File

Some gridding methods (e.g. kriging, radial basis function, minimum curvature, and modified Shepard's method) extrapolate beyond the Z range of the data, which can create inappropriate negative values for parameters like concentration, thickness, and magnitude. The **Grid | Math** command can set negative values equal to zero with the formula $C = \max(A, 0)$.

To set negative values equal to zero:

1. Create a grid file, preferably with a small grid spacing.
2. Click the **Grid | Math** command.
3. In the **Open Grid** dialog, choose the grid file containing unwanted negative values and then click the *Open* button.
4. The grid file appears in *Input Grid File A* in the **Grid Math** dialog.
5. Select an output grid file name in the *Output Grid File C* group by clicking on the  button.
6. In the *Enter a function of the form $C = f(A, B)$* group, enter $C = \max(a, 0)$. This tells the program to return whichever is greater, the value in the grid or zero.

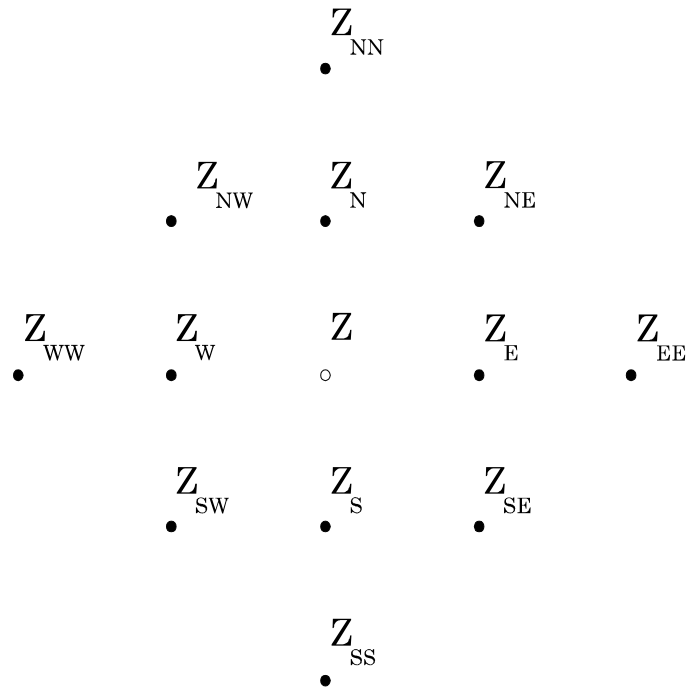
Grid Calculus

The **Grid | Calculus** command provides you with tools to interpret your grid files. Grid calculus can help you define and quantify characteristics in the grid file that might not be obvious by looking at a contour or wireframe of the grid.

The **Grid Calculus** dialog is divided into four sections: *Directional Derivatives*, *Terrain Modeling*, *Differential and Integral Operations*, and *Fourier and Spectral Analysis*. The result of the **Grid Calculus** command is a grid file of the generated data. The generated grid files use the same dimensions as the original grid file but might use different ranges of data depending on the type of output.

When a numerical derivative is needed, central difference formulae are used in the **Grid Calculus** computations in **Surfer**. Because a central difference approach is used, values on both sides of the location for which the derivative is computed are required. This leads to blanking along the edges of the derivative grids (an edge effect).

Surfer uses "compass-based" grid notation to indicate the neighboring grid nodes used for many of the **Grid Calculus** operations, as illustrated below:



***Surfer** uses a compass-based grid notation to indicate the location of the neighboring grid nodes for many of the calculus operations. Node Z in the center of this diagram is the grid node being calculated.*

Chapter 18 - Grid Operations

Using this grid notation, we can write the difference equation approximations for the necessary derivatives at location Z as follows:

$$\frac{dz}{dx} \approx \frac{Z_E - Z_W}{2\Delta x}$$

$$\frac{dz}{dy} \approx \frac{Z_N - Z_S}{2\Delta y}$$

$$\frac{d^2z}{dx^2} \approx \frac{Z_E - 2Z + Z_W}{\Delta x^2}$$

$$\frac{d^2z}{dy^2} \approx \frac{Z_N - 2Z + Z_S}{\Delta y^2}$$

$$\frac{d^2z}{dx dy} \approx \frac{Z_{NE} - Z_{NW} - Z_{SE} + Z_{SW}}{4 \Delta x \Delta y}$$



$$\frac{d^4z}{dx^4} \approx \frac{Z_{WW} - 4Z_W + 6Z - 4Z_E + Z_{EE}}{\Delta x^4}$$

$$\frac{d^4z}{dy^4} \approx \frac{Z_{NN} - 4Z_N + 6Z - 4Z_S + Z_{SS}}{\Delta y^4}$$

$$\frac{d^4z}{dx^2 dy^2} \approx \frac{Z_{NW} - 2Z_N + Z_{NE} - 2Z_W + 4Z - 2Z_E + Z_{SW} - 2Z_S + Z_{SE}}{4 \Delta x \Delta y}$$

Using Grid Calculus

To create a grid using **Grid | Calculus**:

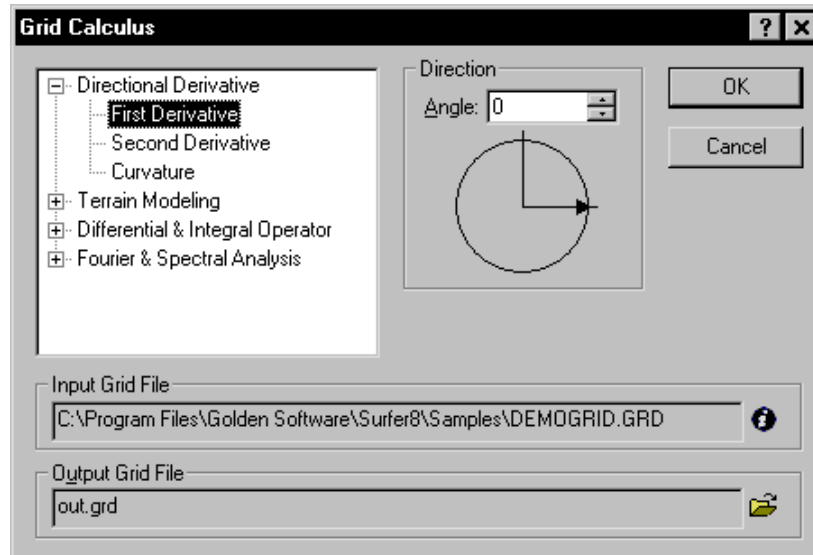
1. Select **Grid | Calculus**.
2. Select a grid file in the **Open Grid** dialog and then click the *Open* button.
3. In the **Grid Calculus** dialog, expand the calculus type you wish to use by clicking on it.
4. Select a calculus operation (e.g. terrain slope).
5. If available, set the options for the operation.
6. Click the  button to display the grid rows, grid columns, XYZ minimums and maximums, and statistics. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.
7. Click the  button to set the output grid path and file name.
8. Click OK and the new grid file is created.

Directional Derivatives

The **Grid | Calculus** directional derivatives provide you with information about the slope, or rate of change of slope, of the gridded surface in a specified direction. Because this takes a specified direction into account, this slope, or rate of change in slope, might not be the steepest slope at a given point. For example, if the specified direction is due East, but the gradient is due North, the directional derivative slope is zero at that point. In the specified direction, there is no slope at that point, although there is a slope to the North.

Directional derivatives are different from the values given in terrain modeling where the direction of the slope is defined as the gradient, or the direction of steepest ascent at a given point (i.e., straight uphill at that point). In the above example, the terrain modeling would report the slope in the north direction at that point. The two methods would report different values at that particular point on the grid.

Let Z be defined as a function of X and Y in a domain that includes point P . At what rate does Z change if we move from point P in a specified direction? In the X axis direction, the rate of change for Z is $\partial Z / \partial x$, and the rate of change in the Y axis direction for Z is $\partial Z / \partial y$.



To use directional derivatives, click on a Directional Derivative on the left, and then set the Angle for the derivative on the right.

Directional Derivative Definition

Let $f(x,y)$ be defined for domain D . Let P be a point of D , let L be a line through P , let Q be a point of L at a distance Δs from P , $\Delta s \neq 0$, such that the line segment PQ lies in D , and let $\Delta f = f(x_Q, y_Q) - f(x_P, y_P)$. Then

$$\lim_{\Delta s \rightarrow 0} \frac{\Delta f}{\Delta s} = \frac{df}{ds}$$

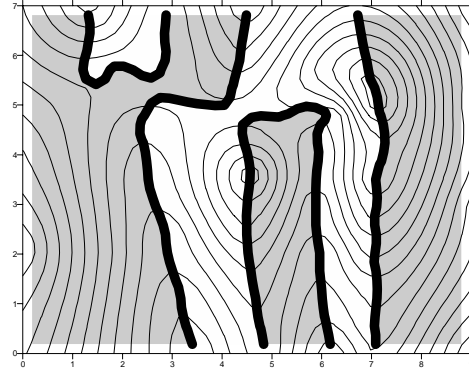
is the directional derivative of $f(x,y)$ at P in the direction of L , that is from P to Q (Schwartz, 1974).

There are three *Directional Derivative* options available: *First Derivative*, *Second Derivative*, and *Curvature*. You can specify the angular direction for the operation.

First Derivative

The **Grid | Calculus** directional derivative *First Derivative* calculates the slope of the surface along the *Direction* line. *First Derivative* grids produce contour maps that show isolines of constant slope along lines of fixed direction. At a particular grid node location, if the slope is uphill, the slope is positive; and if downhill, the slope is negative. Slope is reported as rise over run and can approach negative or positive infinity as the slope approaches vertical in the down or up direction.

The directional first derivative is exactly as in the definition in the *Directional Derivatives* section of this chapter. Instead of a known function $Z(X,Y)$ for which we can compute the necessary limits, **Surfer** uses a grid file so directional derivative is approximated using difference equations.



The original grid file is displayed as a contour map with thin contour lines. The first derivative map is overlaid on the contour map with negative values (downhill in the +x direction) in gray, positive values (uphill in the +x direction) in white, and the zero contour shown as a bold black line.

The directional derivative at a point is equal to the dot product between the gradient vector and a unit vector in the direction of interest:

$$\begin{aligned} \frac{df}{ds} &= \vec{g} \cdot \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix} \\ &= \left[\frac{df}{dx}, \frac{df}{dy} \right] \cdot \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix} \\ &= \frac{df}{dx} \cdot \cos(\alpha) + \frac{df}{dy} \cdot \sin(\alpha) \end{aligned}$$

where α is the user-specified angle (see Schwartz, 1974, p. 785, or Tuma, 1979, p. 89). Using compass-based grid notation, the equation takes the following form:

$$\frac{dZ}{ds} \approx \frac{Z_E - Z_W}{2\Delta x} \cdot \cos(\alpha) + \frac{Z_N - Z_S}{2\Delta y} \cdot \sin(\alpha)$$

Second Derivative

The **Grid | Calculus** directional derivative, *Second Derivative*, calculates the rate of change of slope along a given direction. Second derivative grid files produce contour maps that show isolines of constant rate of change of slope across the surface. If the slope is uphill, the rate of change is positive and downhill is negative.

The directional second derivative is the directional derivative of the first directional derivative:

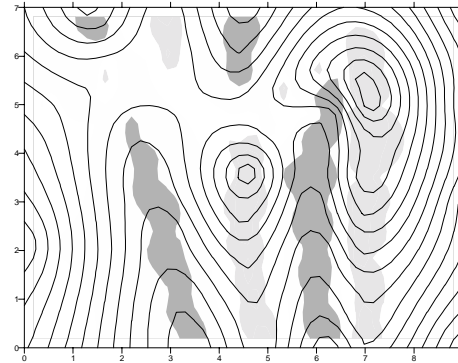
$$\frac{d^2f}{ds^2} = \frac{d\left[\frac{df}{ds}\right]}{ds}$$

$$\frac{d^2f}{ds^2} = \frac{d\left[\frac{df}{dx} \cdot \cos(\alpha) + \frac{df}{dy} \cdot \sin(\alpha)\right]}{ds}$$

$$= \frac{d^2f}{dx^2} \cdot \cos^2(\alpha) + 2 \frac{d^2f}{dxdy} \cdot \cos(\alpha) \cdot \sin(\alpha) + \frac{d^2f}{dy^2} \cdot \sin^2(\alpha)$$

The compass-based grid notation difference equation used in **Surfer** takes the form:

$$\frac{d^2Z}{ds^2} \approx \frac{Z_E - 2Z + Z_W}{\Delta x^2} \cdot \cos^2(\alpha) + \frac{Z_{NE} - Z_{NW} - Z_{SE} + Z_{SW}}{4\Delta x\Delta y} \cdot \cos(\alpha) \cdot \sin(\alpha) + \frac{Z_N - 2Z + Z_S}{\Delta y^2} \cdot \sin^2(\alpha)$$



Contour map of original grid file is overlaid on second derivative map. Areas of greatest change in slope in the X direction occur on ridges (light gray) and valleys (dark gray).

Curvature

The **Grid | Calculus** directional derivative, *Curvature*, is a measure of the rate of change of the inclination angle of tangential planes on a profile line defined by the surface along a specified direction. Curvature is reported as the absolute value of the rate of change and is, therefore, a positive number. *Curvature* is similar to the *Second Derivative*.

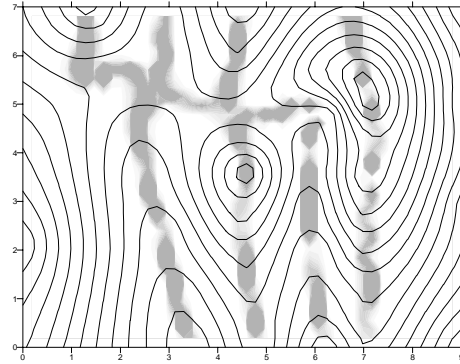
The directional curvature is the absolute value of the rate of change, in a specified direction, of the inclination angle of the tangent plane.

The mathematical formula for the directional curvature of a surface $f(x,y)$ in a direction defined by s is:

$$K_s = \frac{\left| \frac{d^2 f}{ds^2} \right|}{\left[1 + \left(\frac{df}{ds} \right)^2 \right]^{3/2}}$$

(see Schwartz, 1974, p. 592)

$$K_\alpha = \frac{\left| \frac{d^2 f}{dx^2} \cdot \cos^2(\alpha) + 2 \frac{d^2 f}{dxdy} \cdot \cos(\alpha) \cdot \sin(\alpha) + \frac{d^2 f}{dy^2} \cdot \sin^2(\alpha) \right|}{\left[1 + \left(\frac{df}{dx} \cdot \cos(\alpha) + \frac{df}{dy} \cdot \sin(\alpha) \right)^2 \right]^{3/2}}$$

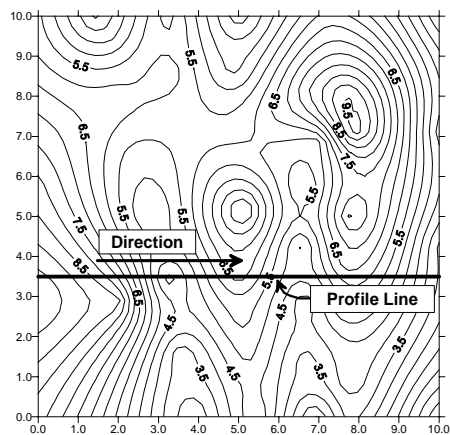


The contour map of the original grid file is overlaid with a color filled directional curvature map. The maximum curvature in the X direction (as indicated by the darker gray contour fill) occurs at valleys and ridges.

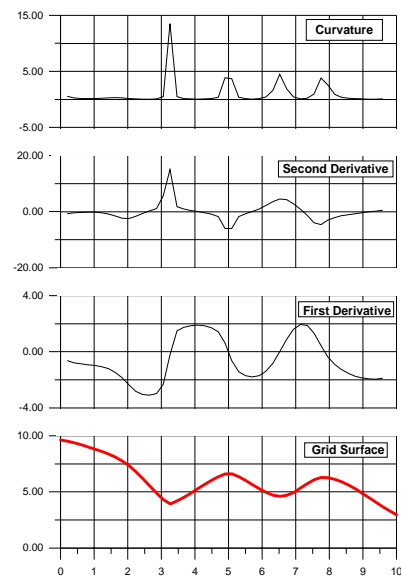
Chapter 18 - Grid Operations

A Graphical Example of Directional Derivatives

The example shown here is based on the DEMOGRID.GRD file included with **Surfer**, although the limits of the grid have been extended to 10 units in both the X and Y dimensions. The map shows the contours and the direction chosen for the **Grid | Calculus** directional derivative operations. A profile line is also included on the map. The graphs included are based on **Grid | Slice** command data collected along the profile lines. Each operation was performed, and the **Slice** command was applied to the resultant grid files. The results are shown in the graphs that were produced in Golden Software, Inc.'s **Grapher**TM. A comparison of the map and the graphs gives a good visual analysis of the results of the directional derivative operations.



This map shows the profile line for the directional derivative and directional curvature calculations.



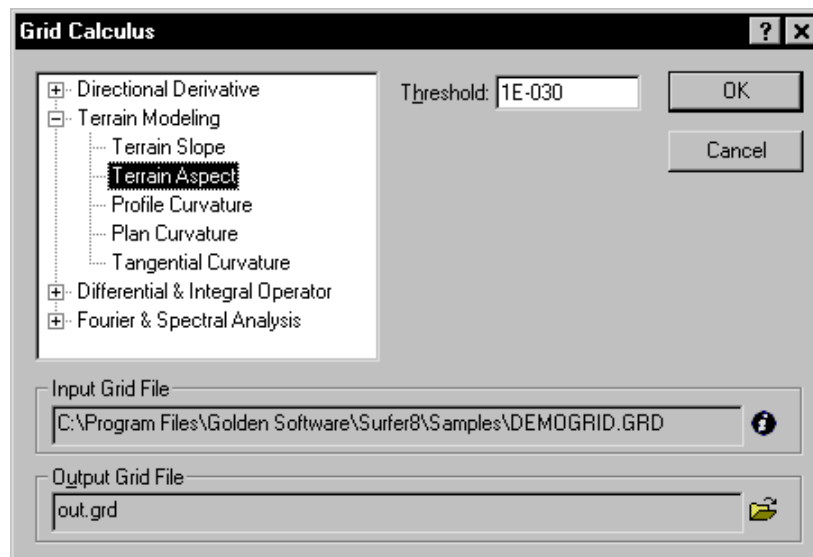
These are cross sectional views of Curvature, Second and First Derivatives, and the original grid surface.

Terrain Modeling

The **Grid | Calculus** terrain modeling option provides additional tools to analyze the surface geometry of a grid file. Results are based on the gradient direction (direction of steepest slope) and not a pre-defined direction as with *Directional Derivatives*. For each grid node location on a surface, **Surfer** determines the magnitude and direction of the steepest slope.

There are five types of *Terrain Modeling* operations: *Terrain Slope*, *Terrain Aspect*, *Profile Curvature*, *Plan Curvature*, and *Tangential Curvature*.

When you choose the *Terrain Aspect*, *Profile Curvature*, *Plan Curvature*, or *Tangential Curvature* options, you can also specify a *Threshold* value. In areas where the steepest slope approaches zero (where the surface is nearly horizontal), the gradient direction is somewhat difficult to ascertain (i.e., the definitions "downhill" and "uphill" are rather arbitrary). In these cases, it is preferable to classify the surface as flat. The *Threshold* value is the minimum value for the slope for which the aspect and curvature values are computed. In locations where the *Threshold* value is not met, the grid node value in the output grid is blanked. The *Threshold* value is, by default, set to a very small value.



To use terrain modeling, click on a model on the left, and then set the Threshold (aspect and curvature) on the right.

Terrain Slope

The **Grid | Calculus** terrain model, *Terrain Slope*, calculates the slope at any grid node on the surface. *Terrain Slope* is reported in degrees from zero (horizontal) to 90 (vertical). For a particular point on the surface, the *Terrain Slope* is based on the direction of steepest descent or ascent at that point (*Terrain Aspect*). This means that across the surface, the gradient direction can change. Grid files of the *Terrain Slope* can produce contour maps that show isolines of constant steepest slope. This operation is similar to the way the *First Directional Derivative* defines the slope at any point on the surface but is more powerful in that it automatically defines the gradient direction at each point on the map.

The slope, S , at a point P is the magnitude of the gradient at that point. From the definition of the gradient:

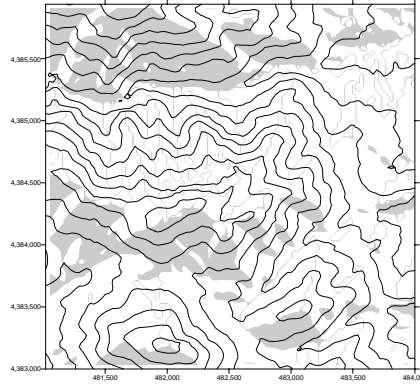
$$S = \sqrt{\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2}$$

Using the compass-based grid notation difference equation yields:

$$S \approx \sqrt{\left(\frac{Z_E - Z_W}{2\Delta x}\right)^2 + \left(\frac{Z_N - Z_S}{2\Delta y}\right)^2}$$

The terrain slope is represented as a slope angle S_T in degrees in keeping with the terrain modeling literature, such Moore et al. (1993):

$$S_T \approx \frac{360}{2\pi} \cdot \arctan \left[\sqrt{\left(\frac{Z_E - Z_W}{2\Delta x}\right)^2 + \left(\frac{Z_N - Z_S}{2\Delta y}\right)^2} \right]$$



In this example, a contour map of the original DEM is overlaid with a terrain aspect map. South-facing slopes (between SE or 135° and SW or 215°) are indicated with gray fills. Gray lines on north facing slopes are contour fills drawn where the aspect varies between 0 degrees (north) and 359 degrees (1 degree west of north).

Terrain Aspect

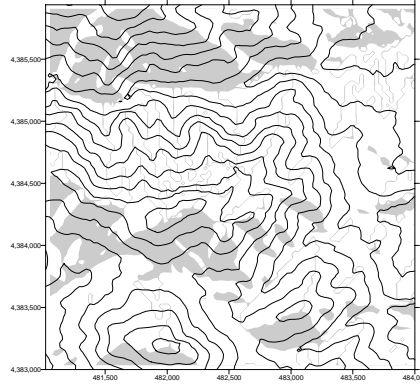
The **Grid | Calculus** terrain model, *Terrain Aspect*, calculates the downhill direction of the steepest slope (i.e. dip direction) at each grid node. It is the direction that is perpendicular to the contour lines on the surface, and is exactly opposite the gradient direction. *Terrain Aspect* values are reported in azimuth, where 0 degrees points due North, and 90 degrees points due East.

In keeping with the terrain modeling literature, e.g. Moore et al. (1993), **Surfer** represents the terrain aspect, A_T , as an azimuth (in degrees, not radians):

$$A_T = 270 - \frac{360}{2\pi} \cdot \text{atan2} \left[\frac{\partial z}{\partial y}, \frac{\partial z}{\partial x} \right]$$

The compass-based grid notation version of this equation is:

$$A_T \approx 270 - \frac{360}{2\pi} \cdot \text{atan2} \left[\frac{Z_N - Z_S}{2\Delta y}, \frac{Z_E - Z_W}{2\Delta x} \right]$$

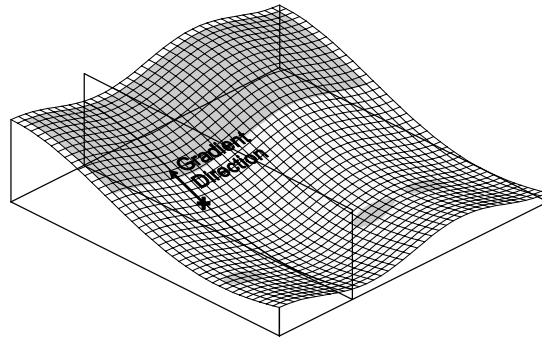


A contour map of the original DEM is overlaid with a terrain aspect map. South-facing slopes (between SE or 135° and SW or 215°) are indicated with gray fills. Gray lines on north facing slopes are contour lines drawn where the aspect varies between 0 degrees (north) and 359 degrees (1 degree west of north).

Profile Curvature

The **Grid | Calculus** terrain model, *Profile Curvature*, determines the downhill or uphill rate of change in slope in the gradient direction (opposite of slope aspect direction) at each grid node. Grid files of *Profile Curvature* produce contour maps that show isolines of constant rate of change of steepest slope across the surface.

This operation is comparable to the *Second Directional Derivative* but is more powerful because it automatically determines the downhill direction at each point on the surface, and then determines the rate of change of slope along that direction at that point. Negative values are convex upward and indicate accelerated flow of water over the surface. Positive values are concave upward and indicate slowed flow over the surface.



Profile Curvature measures the curvature of the surface in the direction of gradient. Negative curvature, shown with a gray fill, indicates a convex upward surface and accelerated water flow.

The profile curvature K_p is given by:

$$K_p = \frac{\left(\frac{\partial^2 z}{\partial x^2}\right)\left(\frac{\partial z}{\partial x}\right)^2 + 2\left(\frac{\partial^2 z}{\partial x \partial y}\right)\left(\frac{\partial z}{\partial x}\right)\left(\frac{\partial z}{\partial y}\right) + \left(\frac{\partial^2 z}{\partial y^2}\right)\left(\frac{\partial z}{\partial y}\right)^2}{pq^{3/2}}$$

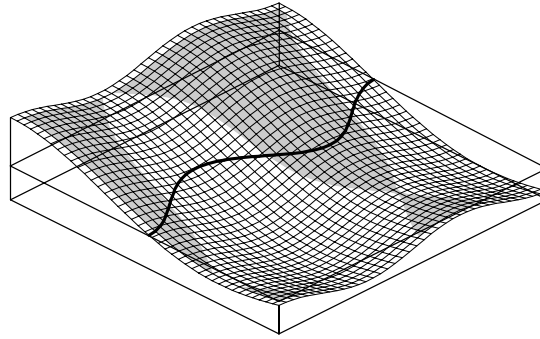
where

$$p = \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2$$

$$q = 1 + p$$

Plan Curvature

The **Grid | Calculus** terrain model, *Plan Curvature*, reflects the rate of change of the *Terrain Aspect* angle measured in the horizontal plane, and is a measure of the curvature of contours. Negative values indicate divergent water flow over the surface, and positive values indicate convergent flow.



***Plan Curvature** calculates the curvature of the surface in the horizontal plane, or the curvature of the contour. Negative curvature, shown with a gray fill, indicates areas of divergent flow.*

The plan curvature K_H is given by:

$$K_H = \frac{\left(\frac{\partial^2 z}{\partial x^2}\right)\left(\frac{\partial z}{\partial y}\right)^2 - 2\left(\frac{\partial^2 z}{\partial x \partial y}\right)\left(\frac{\partial z}{\partial x}\right)\left(\frac{\partial z}{\partial y}\right) + \left(\frac{\partial^2 z}{\partial y^2}\right)\left(\frac{\partial z}{\partial x}\right)^2}{p^{3/2}}$$

where

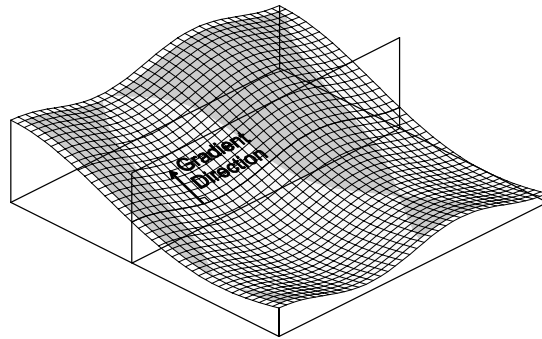
$$p = \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2$$

Tangential Curvature

The **Grid | Calculus** terrain model, *Tangential Curvature*, measures curvature in relation to a vertical plane perpendicular to the gradient direction, or tangential to the contour. The negative and positive areas are the same as for *Plan Curvature*, but the curvature values are different.

Tangential Curvature is related to the *Plan Curvature* K_H by the sine of the slope

$$\gamma: K_H = K_T / \sin \gamma.$$



***Tangential Curvature** measures the curvature of the surface in the vertical plane perpendicular to the gradient direction. Negative curvature, displayed with gray fill, indicates divergent flow conditions.*

Tangential Curvature K_T is given by the equation:

$$K_T = \frac{\left(\frac{\partial^2 z}{\partial x^2}\right)\left(\frac{\partial z}{\partial y}\right)^2 - 2\left(\frac{\partial^2 z}{\partial x \partial y}\right)\left(\frac{\partial z}{\partial x}\right)\left(\frac{\partial z}{\partial y}\right) + \left(\frac{\partial^2 z}{\partial y^2}\right)\left(\frac{\partial z}{\partial x}\right)^2}{pq^{1/2}}$$

where

$$p = \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2$$

$$q = 1 + p$$

Terrain Modeling References

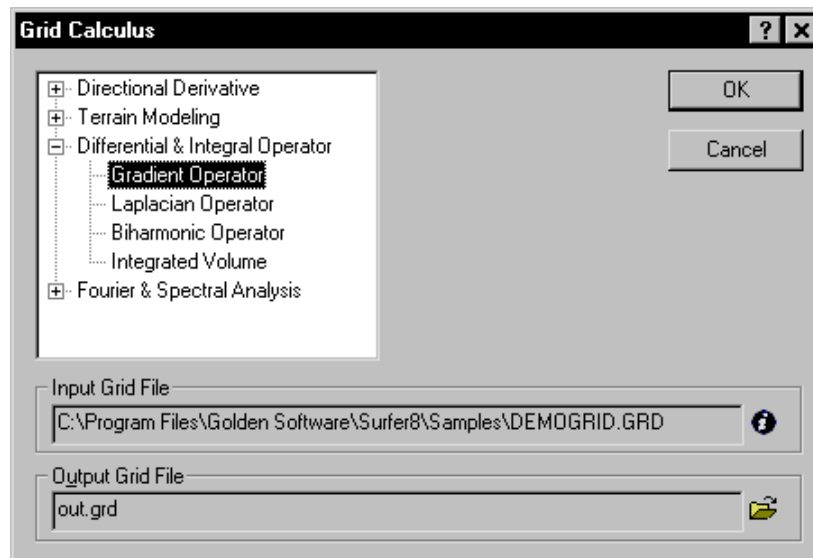
The mathematical definitions used in this implementation were taken from Mitasova and Hofierka (1993). For a general review of the methods and applications of topographic analysis, see Moore et al. (1991).

Mitasova, Helena and Jaroslav Hofierka, (1993), Interpolation by Regularized Spline with Tension: II. Application to Terrain Modeling and Surface Geometry Analysis, *Mathematical Geology*, v. 25, no. 6, p. 657-669.

Moore, I. D., A. Lewis, and J. C. Gallant, (1993), Terrain Attributes: Estimation Methods and Scale Effects, Modeling *Change in Environmental Systems*, A.J. Jakeman et al. editors, John Wiley and Sons, New York.

Differential and Integral Operators

The **Grid | Calculus** differential and integral operators include the *Gradient Operator*, the *Laplacian Operator*, the *Biharmonic Operator*, and *Integrated Volume*.



You can select differential and integral operators from the list on the left side of the dialog. There are no options for Differential & Integral Operators.

Gradient Operator

The *Gradient Operator* generates a grid of steepest slopes (i.e. the magnitude of the gradient) at any point on the surface. This is similar to the *Terrain Slope* operation, but the *Gradient Operator* is reported as a number (rise over run) rather than in degrees, and the direction is opposite that of the slope. The *Gradient Operator* is zero for a horizontal surface, and approaches infinity as the slope approaches vertical.

The definition of the gradient yields the following equation:

$$\|\vec{g}\| = \sqrt{\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2}$$

Using the compass based grid notation yields the equation used in **Surfer**:

$$\|\vec{g}\| \approx \sqrt{\left(\frac{Z_E - Z_W}{2\Delta x}\right)^2 + \left(\frac{Z_N - Z_S}{2\Delta y}\right)^2}$$

Laplacian Operator

The **Grid | Calculus** differential and integral operator, *Laplacian Operator*, provides a measure of discharge or recharge on a surface. In grid files generated with the *Laplacian Operator*, recharge areas are positive, and discharge areas are negative.

Groundwater, heat, and electrical charge are three examples of conservative physical quantities whose local flow rate is proportional to the local gradient. The Laplacian operator, $\nabla^2 Z$, is the mathematical tool that quantifies the net flow into (Laplacian > 0 , or areas of recharge) or out of (Laplacian < 0 , areas of discharge) a local control volume in such physical situations. The Laplacian operator is defined in multivariable calculus by

$$\nabla^2 Z = \frac{\partial^2 Z}{\partial x^2} + \frac{\partial^2 Z}{\partial y^2}$$

The **Surfer** implementation of the *Laplacian Operator* generates a grid using a standard five-point central difference formula.

$$\nabla^2 Z(x,y) = \left(\frac{Z_E - 2Z + Z_W}{\Delta x^2}\right) + \left(\frac{Z_N - 2Z + Z_S}{\Delta y^2}\right)$$

Biharmonic Operator

Bending of thin plates and shells, viscous flow in porous media, and stress functions in linear elasticity are three examples of physical quantities that can be mathematically described using the

Grid | Calculus biharmonic operator. The *Biharmonic Operator*, $\nabla^4 Z$, is defined in multivariable calculus by

$$\nabla^4 Z = \frac{\partial^4 Z}{\partial x^4} + 2 \frac{\partial^4 Z}{\partial x^2 \partial y^2} + \frac{\partial^4 Z}{\partial y^4}$$

This is comparable to applying the *Laplacian Operator* twice.

Integrated Volume

The **Grid | Calculus** differential and integral operator, *Integrated Volume*, gives the accumulated volume over the grid from the southwest to the northeast corner, or over a subgrid from southwest to northeast. This method can calculate volumes over subgrids of any shape. As an example, consider a **Surfer**-generated contour map showing the thickness of a coal seam in a surface coal mine. The *Integrated Volume* map could be used to compute the volume of coal mined in each month. Another example is rainfall intensity for a thunderstorm. If you computed the *Integrated Volume* and then overlaid a watershed boundary map, you could directly compute the quantity of water flowing into each stream.

Mathematically, the integrated volume is defined by

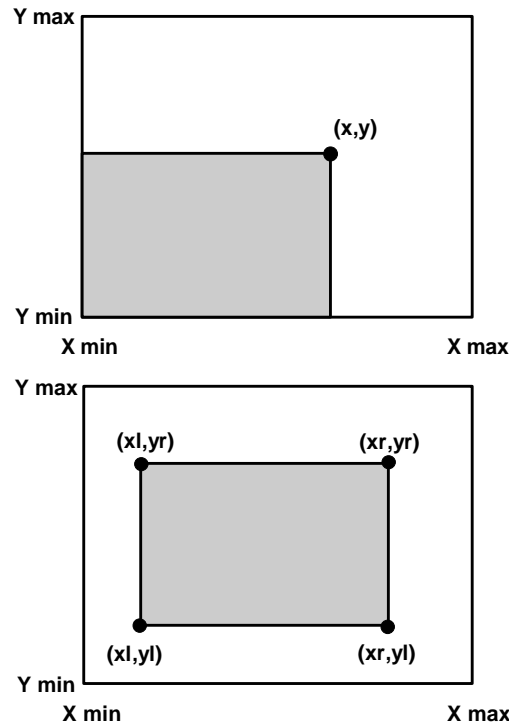
$$V(x, y) = \int_{\alpha=xmin}^x \int_{\beta=ymin}^y z(\alpha, \beta) d\alpha d\beta$$

Chapter 18 - Grid Operations

Consider a surface defined over a rectangular region extending from X min to X max and Y min to Y max:

$V(x,y)$ is then the volume between the plane $Z=0$ and the surface under the shaded area. Thus, $V(xMin,yMin) = 0$, and $V(xMax,yMax)$ equals the volume under the entire rectangular region, which is comparable to the value returned by the **Grid | Volume** command in **Surfer**. If the grid node at $(xMin,yMin)$ is blanked, then the *Integrated Volume* cannot be calculated.

The *Integrated Volume* option enables easy computation of volumes under rectangular subgrids. For example, the volume under the shaded area in the figure at the right is simply $V(xr,yr) - V(xl,yr) - V(xr,yl) + V(xl,yl)$. When applied to a surface specified by a **Surfer** grid, the *Integrated Volume* option generates a dimensionally equivalent grid with the values at grid nodes (x_i,y_i) equal to $V(x_i,y_i)$. You can use the grid node editor to obtain the volume values at different XY locations. The integration is carried out if the surface follows a bilinear functional form over each grid cell.



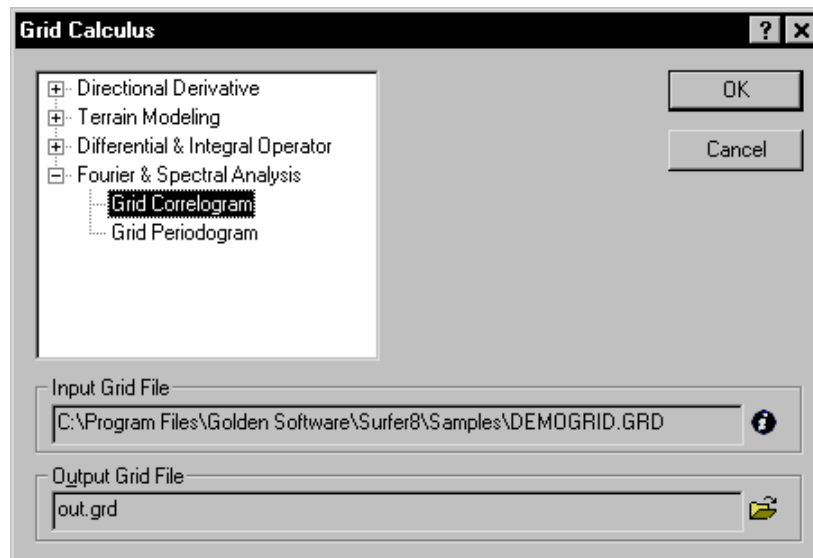
The integral of the bilinear form over the grid cell is the average value of the four corners times the area of the grid cell. Therefore, the underlying formula is

$$V(x_i, y_i) = \frac{Z(x_i, y_i) + Z(x_{i-1}, y_i) + Z(x_i, y_{i-1}) + Z(x_{i-1}, y_{i-1})}{4} +$$

$$V(x_{i-1}, y_i) + V(x_i, y_{i-1}) - V(x_{i-1}, y_{i-1})$$

Fourier and Spectral Analysis

The correlogram and periodogram are used in a variety of fields, including hydrology, agriculture, forestry, meteorology, ecology, and sociology. See, for example, Ripley (1981, Chapter 5), and Bras and Rodriguez-Iturbe (1985, Chapter 6). The correlogram assesses spatial patterns and spatial correlation, while the periodogram identifies periodicities in the data.



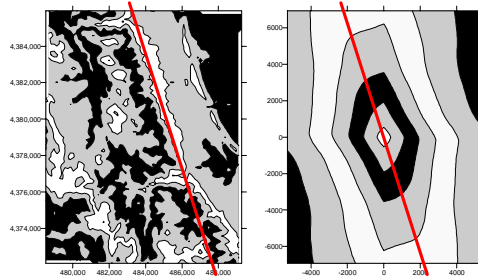
You can select Fourier and spectral analysis operations from the list on the right side of the dialog. There are no options for Fourier & Spectral Analysis.

Chapter 18 - Grid Operations

Correlogram

The **Grid | Calculus** Fourier and spectral analysis, *Correlogram*, assesses spatial patterns and spatial correlation of a grid. Correlograms indicate how well grid values correlate across the grid. They indicate underlying trends in the grid and give a measure of the anisotropy of the grid.

Correlograms are symmetric; $Z(x,y) = Z(-x,-y)$. Grid correlograms have the same dimensions as the parent grids, but the axes represent separation distances.



A contour map of a DEM is shown on the left and the grid correlogram is shown on the right. The heavy line illustrates the approximate direction of anisotropy on both maps.

Given gridded data $z(u,v)$, where $\{u = 0, \dots, NX-1\}$ and $\{v = 0, \dots, NY-1\}$ the grid auto-covariance, $C(r,s)$ is defined as

$$C(r,s) = \frac{1}{NX \cdot NY} \sum_u \sum_v [z(u,v) - \bar{z}][z(u+r, v+s) - \bar{z}]$$

where the summation is over u and v such that

$$\begin{aligned} 0 \leq u \leq NX-1 & \quad 0 \leq u+r \leq NX-1 \\ 0 \leq v \leq NY-1 & \quad 0 \leq v+s \leq NY-1 \end{aligned}$$

and \bar{z} is the arithmetic average of the grid values:

$$\bar{z} = \frac{1}{NX \cdot NY} \sum_u \sum_v z(u,v)$$

Note that the grid auto-covariance function is itself a grid.

The grid auto-covariance is the two-dimensional lattice equivalent of the time series auto-covariance function. It is comparable to the two-dimensional variogram; under the appropriate assumptions of statistical stationarity, there is a one-to-one relationship between the grid auto-covariance function and the variogram. (Beware, however, that the variogram of a raw data set - prior to gridding - is not the same as the variogram of the resulting grid.)

The grid correlogram, $\text{corr}(r,s)$, is a scaled version of the grid auto-covariance function:

$$\text{corr}(r,s) = \frac{C(r,s)}{C(0,0)}$$

$C(0,0)$ is the variance of the data in the grid.

The functions $C(r,s)$ and $\text{corr}(r,s)$ are symmetric in the sense that

$$C(-r,-s) = C(r,s)$$

but,

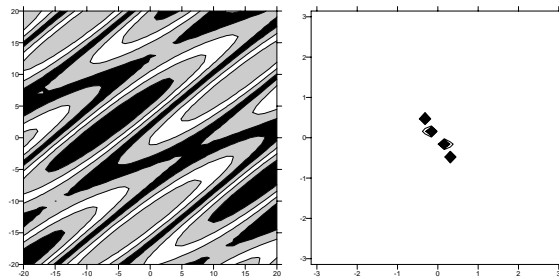
$$C(-r,s) \neq C(r,s)$$

in general. As such, the grids generated by the grid correlogram option of the **Grid | Calculus** sub-system show this symmetry.

Grid Periodogram

The **Grid | Calculus** Fourier and spectral analysis, *Periodogram*, is a decomposition of the surface into the weighted sum of many two-dimensional sinusoids. This operation shows hidden periodicity or repeating patterns that might not otherwise be evident from looking at a contour map or surface.

Surfer computes the grid periodogram for the two-dimensional Fourier frequencies. Grid periodograms have



A contour map of the function $Z = \sin((x/5)-(y/7)) + \cos((y/2)-(x/3))$ is on the left, and the periodogram contour map is on the right. Due to symmetry, the four apparent peaks represent two peaks, one for each of the two fundamental sinusoids.

Chapter 18 - Grid Operations

the same dimensions as the parent grids, but the axes represent the wave numbers associated with the *Fourier* frequencies. The grid periodogram surfaces are symmetric in the sense that $Z(x,y) = Z(-x,-y)$.

In the current implementation of the grid periodogram there is no smoothing, no tapering, and no filtering. Such post-processing can be accomplished using the **Grid | Math** capabilities.

Given gridded data $z(u,v)$, where $\{u = 0, \dots, NX-1\}$ and $\{v = 0, \dots, NY-1\}$ the grid periodogram, $I(\lambda,\mu)$ is formally defined as

$$I(\lambda, \mu) = \frac{N}{4\pi^2} \left| \frac{1}{N} \sum_u \sum_v [z(u, v) - \bar{z}] e^{-i\lambda u} e^{-i\mu v} \right|^2$$

where $i = \sqrt{-1}$.

The value of $I(\lambda,\mu)$ is the weight or strength of the sinusoid with a two dimensional frequency of (λ,μ) .

Breaking this calculation into the component real and imaginary parts yields:

$$A(\lambda, \mu) = \frac{1}{N} \sum_u \sum_v [z(u, v) - \bar{z}] \cos(\lambda u + \mu v)$$

$$B(\lambda, \mu) = \frac{1}{N} \sum_u \sum_v [z(u, v) - \bar{z}] \sin(\lambda u + \mu v)$$

Then

$$I(\lambda, \mu) = \frac{N}{4\pi^2} \cdot [A(\lambda, \mu)^2 + B(\lambda, \mu)^2]$$

where

$$-\pi < \lambda \leq +\pi$$

$$-\pi < \mu \leq +\pi$$

Surfer does not compute the grid periodogram, $I(\lambda, \mu)$, for all possible frequencies (λ, μ) in this range. Rather, it computes the grid periodogram for the two-dimensional *Fourier frequencies*.

A two-dimensional frequency is a Fourier frequency if λ is an integer multiple of $2\pi/NX$, and μ is an integer multiple of $2\pi/NY$. In **Surfer**, we consider the specific frequencies:

$$\lambda = \frac{2\pi j}{NX} \quad j = -\left\lfloor \frac{NX}{2} \right\rfloor, \dots, -1, 0, 1, \dots, \left\lfloor \frac{NX}{2} \right\rfloor$$

$$\mu = \frac{2\pi k}{NY} \quad k = -\left\lfloor \frac{NY}{2} \right\rfloor, \dots, -1, 0, 1, \dots, \left\lfloor \frac{NY}{2} \right\rfloor$$

The indices (j, k) are called the *wave number* of the associated frequency (λ, μ) . **Surfer** follows the engineering convention (rather than the mathematicians' convention) of plotting the grid periodogram as a function of the wave numbers rather than as a function of the frequency: $I(j, k)$. The wave number is defined as:

$$\text{wave number} = 2\pi / \text{wavelength} \quad (\text{e.g. radians/meter})$$

The maximum wave number is the Nyquist frequency $= \pi/\Delta X$ (or Y), which defines the grid limits. Since **Surfer** maintains the grid dimensions (number of columns and rows) from the source grid to the destination grid, there may not be exactly the same number of rows and columns on both sides of the zero lines.

Chapter 18 - Grid Operations

Grid Calculus References

- Akai, Terrence J., (1994), *Applied Numerical Methods for Engineers*, John Wiley and Sons, Inc., New York, 410 pp.
- Bras, R. L., and I. Rodriguez-Iturbe, (1985), *Random Functions and Hydrology*, Addison-Wesley, Reading, Massachusetts, 559 pp.
- Mitasova, Helena and Jaroslav Hofierka, (1993), Interpolation by Regularized Spline with Tension: II. Application to Terrain Modeling and Surface Geometry Analysis, *Mathematical Geology*, v. 25, no. 6, p. 657-669.
- Moore, I. D., R. B. Grayson, and A. R. Ladson, (1991), Digital Terrain Modeling: A Review of Hydrological, Geomorphological and Biological Applications, *Hydro. Proc.*, v. 5, no. 1, p. 3-30.
- Moore, I. D., A. Lewis, and J. C. Gallant, (1993), Terrain Attributes: Estimation Methods and Scale Effects, Modeling *Change in Environmental Systems*, A.J. Jakeman et al. editors, John Wiley and Sons, New York.
- Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, (1992), *Numerical Recipes in C: The Art of Scientific Computing*, Second Edition, Cambridge University Press, New York, 994 pp.
- Ripley, B. D., (1981), *Spatial Statistics*, John Wiley and Sons, New York, 252 pp.
- Schwartz, Abraham, (1974), *Calculus and Analytic Geometry*, 3rd edition, Holt, Rinehart, and Winston, New York, 1140 pp.
- Tuma, Jan J., (1979), *Engineering Mathematics Handbook*, 2nd edition, McGraw-Hill Book Company, New York, 394 pp.

Chapter 19

Volumes, Areas, Cross Sections, and Residuals

Introduction to Volumes, Areas, Cross Sections, and Residuals

The procedures described in this chapter are used to provide more information about the data contained in grid files. The topics discussed in this chapter include volumes, cross sections (slice), and residuals.

The **Grid | Volume** command calculates the volumes and areas for the surfaces defined by grid files or a grid file and a horizontal plane. The **Grid | Volume** computations contains the following information:

- Net volume between two grid files
- Net volume between a grid file and a horizontal plane
- Cut and fill volumes
- Surface area of a grid file above and below a specified level
- Planar area of a grid above and below a specified level

The **Grid | Slice** command outputs a data file that can be used to produce a cross section or profile line. The **Grid | Residuals** command calculates the difference between a grid and the original data. It is also used to calculate the Z value at a specified point.

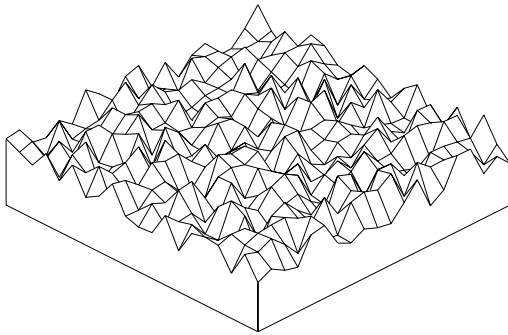
Volumes and Areas

The **Grid | Volume** command is used to compute net volumes, cut and fill volumes, planar areas, and surface areas. The computation results are displayed in the grid volume report. The results can be saved as an ASCII text file, in Rich Text Format, or they can be copied to the clipboard.

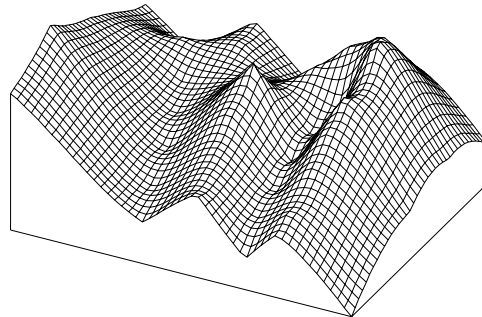
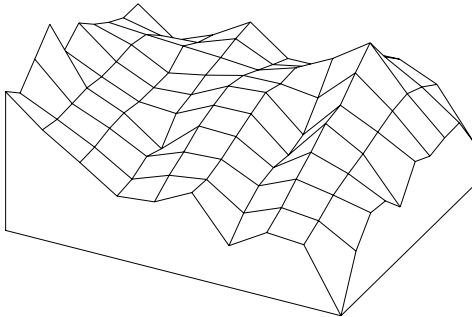
Volume calculations are performed on solids defined by an upper and lower surface. The upper and lower surfaces are defined by a grid file or a plane of constant Z. When two grid files are specified, the grid files must have the same XY limits and the same number of rows and columns. The upper surface does not have to be above the lower surface at all points; the upper surface may dip below the lower surfaces in some locations. Blanked regions on either the upper or the lower surface are excluded from consideration during the volume calculations.

Volume Accuracy

Volume calculations are generated for each grid cell. In areas where the surface is tilted at the top or bottom of a grid cell, **Surfer** approximates the volume of the prism at the top or bottom of the grid cell column. For very coarse grids, the prisms can contain a significant volume. Volume calculations become more accurate as the density of the grid is increased because the relative size of the prisms is reduced compared to the size of the associated column. Grids with significant "noise" or with highly irregular surfaces are poor candidates for volume calculations.



This grid has a highly irregular surface and is a poor candidate for volume calculations.



These two grids are based on the same data but created at different grid densities. The grid on the right is a better candidate for obtaining accurate volume information because it is a much smoother and denser grid.

The Grid Volume Report

When volume computations are performed, the results are displayed in the grid volume report. When the X, Y, and Z units are the same, volume results are given in cubic units. The volume computations in the report includes the following:

Upper Surface and Lower Surface

These sections display the parameters that define the upper and lower surfaces.

Volumes

The volume is calculated by three methods: Trapezoidal Rule, Simpson's Rule, and Simpson's 3/8 Rule. The reported volume is the sum of the *Positive Volume [Cut]* and *Negative Volume [Fill]*. See the *Net Volume* section in this chapter for more information on these methods.

The *Positive Volume [Cut]* is the volume of material in those places where the upper surface is above the lower surface. The *Negative Volume [Fill]* is the volume of material in those places where the upper surface is below the lower surface. The *Net Volume [Cut- Fill]* is the difference between the cut and fill volumes. See the *Cut and Fill Volumes* section of this chapter for more information on cut and fill.

Areas

The *Areas* section reports both planar areas and surface areas. The *Positive Planar Area* represents the planar area where the upper surface is above the lower surface. The *Negative Planar Area* represents the planar area where the upper surface is below the lower surface. The *Blanked Planar Area* is the sum of the areas over the blanked regions on both the upper and lower surfaces. The *Total Planar Area* represents the planar area for the entire grid. Where two surfaces coincide exactly, the area of coincidence is reported as part of the *Positive Planar Area*.

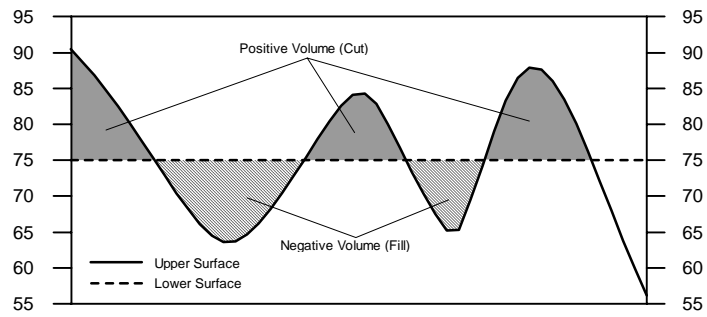
The *Positive Surface Area* represents the area of the surface where the upper surface is above the lower surface. The *Negative Surface Area* represents the area of the surface where the upper surface is below the lower surface.

Net Volume

The volume calculation determines the net volume between the upper and lower surface. The net volume is reported in the *Volumes* section of the grid volume report.

Cut and Fill Volumes

To visualize net volume, consider a construction site where the topography must be graded to a flat surface prior to the beginning of construction. The upper surface represents the current topography, and the lower surface represents the final graded site elevations. In some places, cut must be made into the current topography to remove earth to the level of the final site. In other areas, earth may be needed to fill in areas where the current topography is below the elevation of the final site. The net volume is the difference between all cuts and all fills. If the volume is positive, earth needs to be removed from the site to achieve the final level. If the volume is negative, earth needs to be hauled into the site to achieve the final planned grade for the site.



This is an example of a cross-section showing the relation between the upper and lower surfaces, and the cut and fill volumes. The lower surface is defined by $Z = 75$. The map is shown below.

Blanked regions on either the upper or lower surface are excluded from consideration during the cut and fill calculations.

Volume Calculations

Three methods are used to determine volumes: Trapezoidal Rule, Simpson's Rule, and Simpson's 3/8 Rule. The difference in the volume calculations by the three different methods measures the accuracy of the volume calculations. If the three volume calculations are reasonably close together, the true volume is close to these values. If the three values differ somewhat, a new denser grid file should be used before performing the volume calculations again. The net volume can be reported as the average of the three values.

Mathematically, the volume under a function $f(x, y)$ is defined by a double integral

$$\text{Volume} = \int_{x_{\min}}^{x_{\max}} \int_{y_{\min}}^{y_{\max}} f(x, y) dx dy$$

In **Surfer**, this is computed by first integrating over X (the columns) to get the areas under the individual rows, and then integrating over Y (the rows) to get the final volume. This approach is explained in detail in Press, et al., 1988, Section [4.6].

Surfer approximates the necessary one-dimensional integrals using three classical numerical integration algorithms: extended trapezoidal rule, extended Simpson's rule, and extended Simpson's 3/8 rule; see Press et al., 1988, Section [4.1]. In the following formula, Δx represents the grid column spacing, Δy represents the grid row spacing, and $G_{i,j}$ represents the grid node value in row i and column j .

Extended Trapezoidal Rule

The pattern of the coefficients is $\{1, 2, 2, 2, \dots, 2, 2, 1\}$:

$$A_i = \frac{\Delta x}{2} [G_{i,1} + 2G_{i,2} + 2G_{i,3} + \dots + 2G_{i,nCol-1} + G_{i,nCol}]$$

$$\text{Volume} \approx \frac{\Delta y}{2} [A_1 + 2A_2 + 2A_3 + \dots + 2A_{nCol-1} + A_{nCol}]$$

Extended Simpson's Rule

The pattern of the coefficients is $\{1, 4, 2, 4, 2, 4, 2, \dots, 4, 2, 1\}$:

$$A_i = \frac{\Delta x}{3} [G_{i,1} + 4G_{i,2} + 2G_{i,3} + 4G_{i,4} + \dots + 2G_{i,nCol-1} + G_{i,nCol}]$$

$$\text{Volume} \approx \frac{\Delta y}{3} [A_1 + 4A_2 + 2A_3 + 4A_4 + \dots + 2A_{nCol-1} + A_{nCol}]$$

Extended Simpson's 3/8 Rule

The pattern of the coefficients is $\{1, 3, 3, 3, 2, 3, 3, 2, \dots, 3, 3, 2, 1\}$:

$$A_i = \frac{3\Delta x}{8} [G_{i,1} + 3G_{i,2} + 3G_{i,3} + 2G_{i,4} + \dots + 2G_{i,nCol-1} + G_{i,nCol}]$$

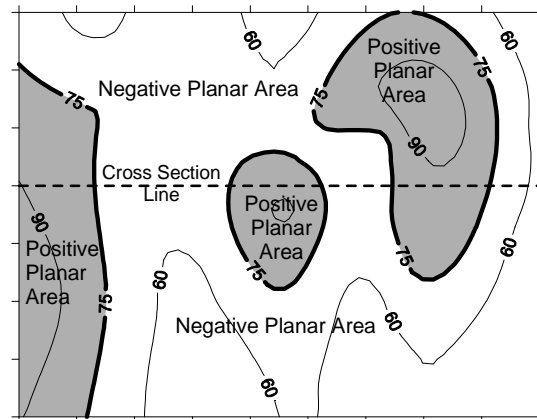
$$\text{Volume} \approx \frac{3\Delta y}{8} [A_1 + 3A_2 + 3A_3 + 2A_4 + \dots + 2A_{nCol-1} + A_{nCol}]$$

Volume results are provided in cubic units based on the units of the input grid file. For a grid with XYZ units in feet, the units for the results are: Net Volume = (feet * feet * feet)

Blanked regions on either the upper or the lower surface are excluded from consideration during the volume calculations.

Planar Area Calculations

Planar area is computed by projecting the cut and fill portions of the surface onto a plane and calculating the area of the projection. *Positive Planar Area* represents the projection of the cut (map areas where the upper surface is above the lower surface) onto a horizontal plane. *Negative Planar Area* represents the projection of the fill (map areas where the upper surface is below the lower surface) onto a horizontal plane.



This is a contour map showing the relation between positive and negative planar area on a map. The lower surface is defined by a plane where Z = 75.

Surface Area Calculations

The surface area is the actual area of the surface. *Positive Surface Area* corresponds to the area of the surface where the upper surface is above the lower surface. *Negative Surface Area* corresponds to the area of the surface where the upper surface is below the lower surface.



As an example, consider a grid file representing the bottom of a pond, with the water level at a Z value of zero. If $Z = 0$ is defined as the upper surface and the grid representing the pond bottom is defined as the lower surface, you can determine the surface area of the bottom of the pond. Suppose you want to drain the pond and line it with plastic. You need to know the area of the plastic necessary to line the pond bottom. In this case, the negative surface area represents the surface of the pond bottom. This would tell us precisely the amount of plastic necessary to line the pond bottom.

Volume Reference

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. (1988), *Numerical Recipes in C*, Cambridge University Press.

Calculating Volumes and Areas

Use the **Grid | Volume** command to calculate net volumes, cut and fill volumes, planar areas, and surface areas.

1. Click the **Grid | Volume** command.
2. In the **Open Grid** dialog, select the grid file to use in the volume and area calculations, and then click the *Open* button. This can be the grid file for either the upper or the lower surface.
3. The **Grid Volume** dialog is displayed with the specified grid file shown for both the upper and lower surface.
4. Specify the *Upper Surface* and *Lower Surface* parameters.
 - The  button displays information about the grid or DEM file used in the volume calculations. The information includes the grid size, the minimum and maximum XYZ values contained in the grid file, and statistics. If the grid file is large, click OK in the message box that appears to create a detailed grid report or click *Cancel* to create a shorter, less detailed grid report.
 - The *Grid File* option is used to specify a grid to use as the upper or lower surface. To use the grid file for the surface, click the *Grid File* option. To change the grid file for either the upper or lower surface, click the *Grid File* option and then click the  button to select another grid file.

Chapter 19 - Volumes, Areas, Cross Sections, and Residuals

- The *Constant* option is used to specify the level of the horizontal planar surface to use as the upper or lower surface. Type horizontal planar surface value, in data units, into the *Z =* box.
5. If the XY and Z units are different, the resulting volume calculations are meaningless. If, for example, your XY values are in meters but your Z values are in feet, the volume results are square meters times feet, rather than cubic units. *Z Scale Factor* can be used to adjust for this problem. In this example, setting the *Z Scale Factor* to 0.3048 (number of meters in a foot) results in cubic meters for volume calculations. You will need to know the conversion factor to use this field.
 6. Click OK in the **Grid Volume** dialog and the results are displayed in a report.
 7. Use **File | Save As** to save the report, or you can copy the information to the clipboard and paste it into another application.




*The Upper Surface or Lower Surface can be a constant or a grid file in the **Grid Volume** dialog.*

Calculating the Volume between Two Surfaces

A common problem solved with a volume calculation is determining the volume of material between two grid files. For example, you might have drill hole data that provides the elevation of the top and bottom of a coal seam, and you want to determine the volume of coal present in the seam. The elevation of the top and bottom of the coal seam defines the upper and lower surfaces, respectively. In this case, the lower surface would be entirely below the upper surface.

To find the volume between two surfaces:

1. Create grid files of the upper and lower surfaces. In the example here, the grid files would define the elevation of the top and bottom of the coal seam. Both grids must use the same dimensions and the same number of rows and columns.
2. Choose **Grid | Volume** and select the upper surface grid file in the **Open Grid** dialog and then click the *Open* button.
3. In the **Grid Volume** dialog, the *Grid File* option button should be selected in the *Upper Surface* group.
4. In the *Lower Surface* group, click the *Grid File* option button and then click the  button.
5. In the **Open Grid** dialog, select the lower surface grid file and then click the *Open* button.
6. In the **Grid Volume** dialog, verify that the correct grid files are specified for the upper and lower surfaces and click OK. The grid volume report is generated. The volume between the surfaces is reported as the *Positive Volume [Cut]*, which, in this case, is the same as the *Volume* calculation.
7. To save the volume information, use the **File | Save As** command in the report window and specify a name and file extension. Click *Save* and the information is saved.

Calculating the Volume of a Lake

To illustrate the capability of using the **Grid | Volume** command, consider the application of calculating the volume of water within a lake. This application requires that a grid file be created which defines the lake bottom. The other surface is defined as a horizontal planar surface that defines the water surface of the lake.

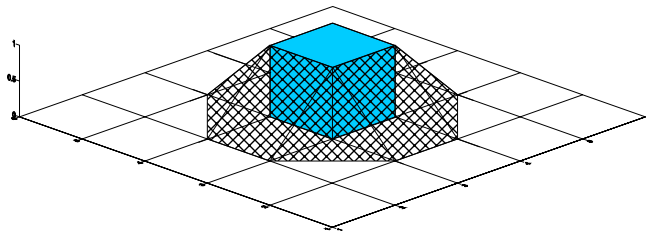
To calculate the volume of a lake:

1. Create the grid file for the lake. If the lake bottom is defined as the depth of the water, use negative Z values so that the lake appears as a basin rather than a hill. It is also recommended that data points be added beyond the limits of the lakeshore so that the surface is above the level of the lake in those areas.
2. Select **Grid | Volume**.
3. Select the grid file for the lake in the **Open Grid** dialog and then click the *Open* button.
4. In the **Grid Volume** dialog, click the *Grid File* option in the *Lower Surface* group.
5. In the *Upper Surface* group, click the *Constant* option. Enter the value for the lake surface into the Z = box. For example, if you are using depth in your grid file as a negative value, the Z = value for your horizontal (lake) surface should be zero.
6. Click OK in the **Grid Volume** dialog and the volume and area calculations are reported. The volume of the lake is reported as the *Positive Volume*.
7. To save the information, use the **File | Save As** command in the grid volume report window and specify a name and file extension.

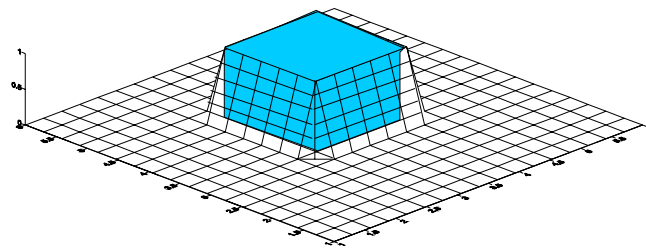
Volume of a Blanked Grid

Surfer can be used to calculate the volume between two grid surfaces or between a grid surface and a horizontal plane. The accuracy of the volume calculation increases as the number of grid lines in the grid file increases.

Consider the way **Surfer** calculates the volume of a cube that is 1 foot x 1 foot x 1 foot. **Surfer** cannot generate a surface that is a perfect cube since each XY coordinate can only have a single z value. A diagonal line is drawn from the top surface to the lower with a width of one grid unit. Using a grid with one-foot grid line spacing, the resulting surface looks like a pyramid with the top chopped off.



Volume error is increased in a cube with a course grid.

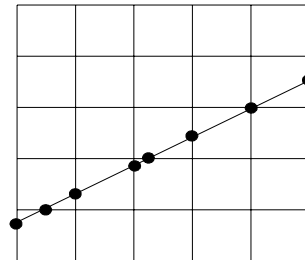


Volume error is reduced in a cube with a finer grid.

Cross Sections

The **Grid | Slice** command produces cross section data. The cross section is generated by taking a vertical slice through a gridded surface along a boundary line. Surface traces on a cross section are sometimes referred to as profile lines.

The surface is based on a grid file and the boundary line is based on the information in a blanking file [.BLN]. See *Appendix C, File Formats* for more information on creating blanking files.



At each point where the boundary line crosses a grid line, a cross section data point is generated.

Cross Section Data File

Cross section data are written to an ASCII data file [.DAT] during the **Slice** operation. The ASCII data file consists of five columns of data. When the profile is written to the ASCII file, a new row of data is generated for each point where the boundary line intersects the grid lines in the data file. The columns in the data file are arranged as follows:

Column	Contents
Column A	X coordinate of the boundary line and grid line (row or column) intersection
Column B	Y coordinate of the boundary line and grid line (row or column) intersection
Column C	Z value at the boundary line and grid line (row or column) intersection
Column D	Column D contains the accumulated horizontal distance along the boundary line. This distance is accumulated even if the boundary lines go outside the grid region. The distance is reset to zero for each new boundary in the blanking file.
Column E	Column E contains the boundary number, used when more than one boundary line is contained in the file. Each new boundary line in the blanking file is assigned a different number.

Cross Section Blanking File

A blanking file may also be produced with the slicing operation. The blanking file contains three columns of data:



Column	Contents
Column A	X coordinate at each point where the boundary intersects a grid line
Column B	Y coordinate at each point where the boundary intersects a grid line
Column C	Z value at each point where the boundary intersects a grid line

Blanking Files

Blanking files are used to create cross sections, to blank grids, or to show boundary information on a map. Cross sections use blanking files to determine the cross sectional data. Blanking files can be created in an ASCII editor, or they can be created using the **Map | Digitize** command. See *Creating a Blanking File with the Digitize Command* in *Chapter 7, Base Maps* for information on creating a blanking file with **Map | Digitize**. If you know the coordinates of the cross section line and wish to create a blanking file in an ASCII editor, refer to the blanking file format in *Appendix C, File Formats*.

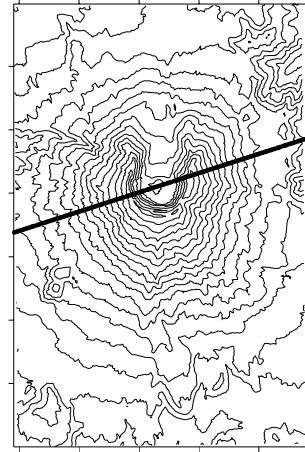
Creating Cross Section Data

To create a data file containing cross section data:

1. Select the **Grid | Slice** command.
2. In the **Open Grid** dialog, select a grid file and then click the *Open* button.
3. In the **Open** dialog, select the blanking file to use when producing the profile data and then click the *Open* button.
4. If desired, enter a name for the *Output BLN File* in the **Grid Slice** dialog by clicking the  button to open the **Save As** dialog.
5. Specify the name for the *Output DAT File* by clicking the  button to open the **Save As** dialog.
6. The *Set values outside grid to* option assigns a special Z value to points that lie beyond the grid limits when the check box is active. If this option is not activated, points outside the grid limit are not included in the output data file. The default value is equal to the following equation:

$$(1.4 \times \text{Grid Z Minimum}) - (0.4 \times \text{Grid Z Maximum})$$

7. The *Set values in blanked grids to* option assigns a special Z value to points that lie inside blanked regions when the check box is active. If this option is not activated, points inside the blanked areas are not included in the output data file. The equation for the default value is given in step 6.
8. Click OK and a data file (and a blanking file, if specified) is created.



You can display the profile line across the contour map. This is the profile line used to produce the following cross section.

Plotting Cross Sections

Cross Section Line

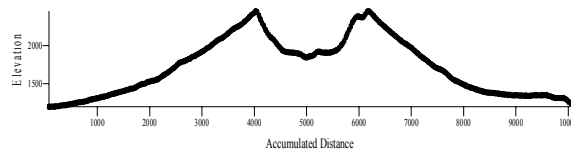
To display the location of the slice for the cross section on a map, use the **Map | Base Map** command. See *Chapter 7, Base Maps* for more information about base maps.

Cross Section Data

In **Surfer**, you can use the **Map | Post Map** command to make a quick plot of the cross section projected onto a specific plane. To create more elaborate cross sections, we advise using Golden Software's **Grapher** program.

To plot the cross section:

1. Select **Map | Post Map | New Post Map**.
2. Chose the cross sectional data file in the **Open** dialog, and then click the *Open* button.
3. Double-click on the word "Post" in the **Object Manager** to open the post map properties.
4. Choose the columns of data you wish to plot (see page 448 for the data file contents).
 - Using Column D (accumulated distance) as the X coordinate and Column C (Z value) as the Y coordinate produces a profile line that shows the true accumulated distance along the boundary line.



*This is the cross section taken from the profile line on the map. The cross section was created as a **Post Map** using Column D as the X coordinate and Column C as the Y coordinate.*

- Plotting Column A (X coordinate) as the X coordinate and Column C (Z value) as the Y coordinate produces a profile line projected onto the X-Z plane.
 - Using Column B (Y coordinate) as the X coordinate and Column C (Z value) as the Y coordinate produces a profile line projected onto the Y-Z plane.
5. Choose a symbol style, color, and size to create your profile. If your grid is sparse, you will see "points" rather than a "line" on the plot. Sometimes this can be altered by choosing larger symbols.

Residuals

The **Grid | Residuals** command computes the vertical difference between the Z value in the data file and the interpolated Z value on a gridded surface. It gives a quantitative measure of how well the grid file agrees with the original data. The command can also be used to obtain the Z value of the surface at any point.

A residual is the difference between the Z value of a point in a data file and the interpolated Z value at the same XY location on a gridded surface. Residual values are reported as either positive or negative values. If the Z value in the data file is greater than the Z value derived from the gridded surface, the residual value is positive. A negative residual value indicates that the Z value from the data file is less than the interpolated Z value at that point. If the grid file is blanked at the data point, the residual value is left blank in the worksheet.

The formula used to compute a residual value is:

$$Z_{res} = Z_{dat} - Z_{grd}$$

where

Z_{res} = the residual value

Z_{dat} = the Z value in the data file

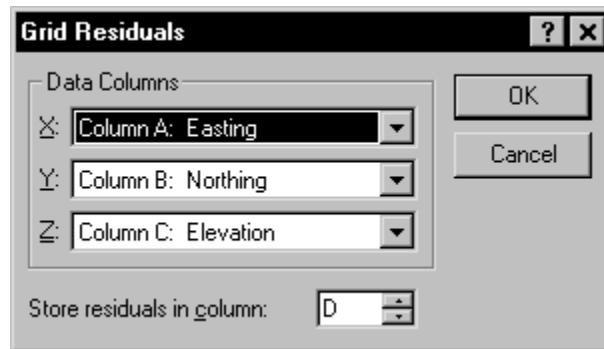
Z_{grd} = the Z value of the surface at the XY coordinate from the grid file

A worksheet window is automatically opened when the **Residuals** command is performed. The worksheet includes the data file used for the calculation and the newly generated residual information. Use the worksheet's **Data | Statistics** command to obtain statistical information on the calculated residuals. See *Chapter 3, Data Files and the Worksheet* for more information about the worksheet and worksheet commands.

Calculating Residuals

To calculate residuals:

1. Choose the **Grid | Residuals** command.
2. In the **Open Grid** dialog, select the grid file to be used for the residual calculation and then click the *Open* button.
3. In the **Open** dialog, select the data file to be compared to the grid file, and then click the *Open* button.
4. In the **Grid Residuals** dialog, specify the location for the X, Y, and Z coordinates in the *Data Columns* group.
5. In the *Store residuals in column* box, enter the column where you would like the residuals stored. By default, **Surfer** selects the first empty column in the data file.
6. Click OK and the worksheet window opens. The worksheet window displays the data file as it was specified in the **Grid Residuals** dialog with the additional column for the residual data.



*Specify the locations for the X, Y, and Z data and the location for the calculated residuals in the **Grid Residuals** dialog.*

DEMOGRID.DAT				
	D2		0.24485219319662	
	A	B	C	D
1	Easting	Northing	Elevation	Residuals
2	0.1	0	90	0.2448522
3	3.5	0	45	-0.2151875
4	4.9	0	65	1.2410420
5	6.2	0	40	-1.2948816
6	7	0	55	0.4060126

*After clicking OK in the **Grid Residuals** dialog, the worksheet window opens with the original data and the new residuals column.*

Calculating the Value of the Surface at a Point

The **Grid | Residuals** command allows you to obtain the Z value of a grid at any point. To perform the operation, an XYZ data file must be created that contains the XY coordinates for the point you want to calculate. When creating the file, specify a value of "0" for the Z coordinate.

For example, consider a grid file that ranges from zero to 10 in the X and Y dimensions, and you want to determine the value of the surface at the point (1,3) on the surface. If an XYZ data file containing the coordinate (1,3,0) is created, you can use the **Residuals** command to determine the value of the surface at this point. The point does not need to have the same coordinates as a grid node.

To calculate the grid value at a point:

1. Open a new worksheet window with **File | New**. Refer to *Chapter 3, Data Files and the Worksheet* for more information on using the worksheet.
2. In Column A, Row 1, enter the X coordinate for the point at which you want to determine the Z value. In Column B, type the Y coordinate, and in Column C, type the value zero. Add as many points as you want into the data file by entering the coordinates for each point into a different row in the data file.
3. Use the **File | Save As** command and enter a name for the data file, for example RESID.DAT.
4. Click *Save*.
5. Click OK in the **GSI Export Options** dialog.
6. Close the worksheet.
7. Open a plot window.
8. Choose the **Grid | Residuals** command.
9. Select the grid file in the **Open Grid** dialog and then click *Open*.
10. Select the data file you just created, for example RESID.DAT, in the **Open** dialog and then click *Open*.
11. In the **Grid Residuals** dialog, specify the columns for X, Y, and Z. Also, specify the column to which the residual values are written in the worksheet.
12. Click OK and the worksheet window is opened with the data file including the column with the residual values.
13. Use the **Data | Transform** command and multiply the reported residual values by negative one (-1) to get the actual Z value of the surface at the points specified in the data file.

Chapter 20

Importing and Exporting Maps and Graphics

Introduction to Importing and Exporting

Surfer supports several file formats to aid in the exchange of maps and graphics with other applications. Files from other applications can be imported into **Surfer** as stand-alone graphics or as a base map. For example, you can export files from **Surfer** as metafiles and then import them into other software. The clipboard provides another easy way to move maps to other Windows applications.

Importing Objects into Surfer

There are three ways you can import drawing objects into **Surfer**. You can use the **File | Import** command to import objects such as polylines, polygons, and points as stand-alone graphics. You can use the **Map | Base Map** command to import objects as a base map. Finally, you can use the clipboard to copy objects and paste them into a plot window.

Import vs. Base Map

The **File | Import** command is similar to the **Map | Base Map** command except that the file is imported as a composite graphic object rather than as a map. Use **File | Import** to import graphic "trim" objects, background graphics, or some form of annotation. Use **Map | Base Map** to import boundary information suitable for overlaying on top of other map types. The Base Map command is discussed in *Chapter 7, Base Maps*.


Import

The composite object created by **File | Import** is made up of two or more objects grouped together into a single object. The **Arrange | Break Apart** command can be used to disassociate the composite object into its separate parts. For example, when importing a file containing several polygons, the file is initially imported as a single composite object composed of the polygons. When you break apart the composite, each polygon becomes a separate entity. Each polygon can then be modified individually. Metafiles are created as composite objects, as well. When a metafile is broken apart, different properties can be applied to the same type of objects. For example, each line could employ unique properties.

Chapter 20 - Importing and Exporting Maps and Graphics

If the imported file contains a single object, it is not imported as a composite object. For example, if a bitmap is imported, only a single bitmap is created. Alternatively, if the file contains a single polyline, the file is imported as a single polyline and not as a composite.

Import Options

Import options for specific file types are discussed in the online help. Click the  button in the import dialog for more information about an import option.

Paste Special

An alternative to **Import** or **Base Map** is pasting. The **Edit | Paste Special** command is similar to **Edit | Paste**, but it allows you to specify which clipboard format to use when pasting. Use the **Edit | Paste Special** command for selecting one of the clipboard formats: *GS_Surfer2*, *Enhanced Metafile*, *Picture*, *Device Independent Bitmap*, and *Text*.

- The *GS_Surfer2* format lets you cut or copy an entire **Surfer** map or drawing to the clipboard. It is stored in the clipboard as a native **Surfer** object. You can paste this map or drawing into another plot window in **Surfer**. The map is identical to the original and can be edited. This is an easy way to place many complete objects on a single page.
- The *Enhanced Metafile* and *Picture* formats are Windows metafile formats, which store objects as a series of Windows drawing commands. Metafiles can be resized without distorting the image. These formats are supported by most Windows applications. When you select these formats, you can activate the *Break Apart Metafiles* check box in the **Clipboard Formats** dialog. This option converts the metafile into a series of native **Surfer** objects and groups them as a single composite object. Composite objects can be broken apart using the **Arrange | Break Apart** command. For example, you could paste a contour map in the picture format, and activate the *Break Apart Metafiles* check box. In this case, the contour map is pasted as a group of polylines and text. You can then break apart the contour map and modify the individual polylines (such as the contour lines).
- The *Device Independent Bitmap* format stores objects as bitmaps. Bitmaps are difficult to resize without distorting the image, and their colors are limited. This format is relatively common, and is supported by most other Windows applications.
- The *Text* format is used to import text. Imported text can contain any number of lines, and uses the default text properties. If you want to paste both text and other objects, and retain the text, you must copy and paste the text separately from the other objects.


Exporting Objects

The **File | Export** command lets you save information in files for use in **Surfer** and other programs.

When exporting, you can specify the objects to include in the exported file. Normally, all objects are exported. However, if you check the *Selected objects only* option in the **Export** dialog, then only selected objects are exported to the file.

All export formats have limitations in what they can represent. The only way to preserve all information within **Surfer** is to save as a **Surfer** file [.SRF] using the **File | Save** or **File | Save As** command. These files can only be read in **Surfer 8**.

Export Options

The export options are self-explanatory in most instances. Where additional information is required, you can click the  button in the file type **Export Options** dialog.

Specifying File Type During Export

When exporting to a particular file type, you can specify the type of file by typing the appropriate file extension after the file name in the *File name* box. For example, if you want to save your map as a [.DXF] file, you can type MYMAP.DXF into the *File name* box. The [.DXF] extension indicates to **Surfer** that you want to save the map in an AutoCAD DXF format. Otherwise, you can choose the type of file from the *Save as type* list. In this case, it is not necessary to specify the file extension, as **Surfer** adds it automatically.

Bitmap Export

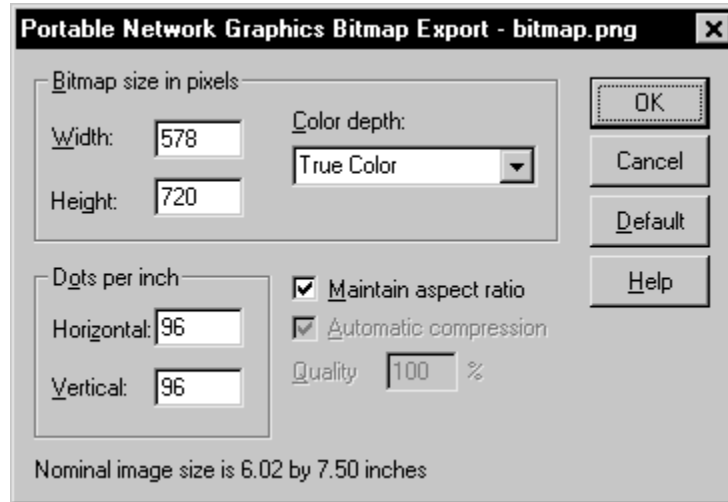
Bitmap export formats let you specify the bitmap size, in pixels or dots per inch, for the exported file. When a bitmap file type is selected, a dialog appears prompting for the dimensions to use for the file.

Bitmap Size

One method of exporting bitmaps is to use the *Bitmap size in pixels* options. You can use the *Width* and *Height* values, which are in proportion to the dimensions of the image you are creating. To maintain an aspect ratio of 1:1 in the output image, check *Maintain aspect ratio*. The dimensions of the image are listed in the *Nominal image size* statement at the bottom of the bitmap export dialog. As an example, if the page coordinates are in inches, you could multiply these values by 100 and use them for the *Width* and *Height* options. This provides a proportionally

scaled bitmap resolution of 100 pixels per inch. (Note: If using centimeters, you should use values such as 25 pixels per centimeter to avoid extremely large bitmap files.) If you know how many dots per inch you wish to have in the exported bitmap, set them in the *Horizontal* and *Vertical* boxes in the *Dots per inch* group.

Setting the *Bitmap size in pixels* changes the *Dots per inch* values and vice versa. Click the *Default* button to set all controls to their default values.



You can set the bitmap size in dots per inch or in pixel dimensions in the bitmap export dialog.

Color Depth

You can choose a color depth from the *Color depth* drop-down list for the output bitmap. Greater color depth provides a greater fidelity of the colors, but the file size increases. Different bitmap formats support different color depths. Some output formats support 256 colors only, while other formats also support true color. The four possible color depths are monochrome, 16 colors, 256 colors, and true color.

JPEG Export

There are extra options when exporting to a JPEG file. If the *Automatic compression* check box is checked, the image is exported using a compression ratio the program determines as most appropriate. If the *Automatic compression* check box is not checked, specify what quality (0 - 100) you would like the image to be exported in the *Quality* box. Specifying 0 yields the most compression, but the lowest quality. Specifying 100 yields the highest quality, but the least compression.

Coordinates and Scaling During Export

Some export types can use either page coordinates or map coordinates in the exported file. Page coordinates specify the position of the objects as they appear on the printed page. Map coordinates are the XY coordinates used by the map as read from the grid file or data file used to produce the map. By default, **Surfer** transforms page coordinates into map coordinates when exporting a single map. This assures that your map coordinates are used in the file.

If there are objects other than maps in the window, such as drawn polylines or polygons, the export coordinates default to page coordinates.

In the **Export** dialog, page coordinates are specified in the *Page Rectangle* boxes. Map coordinates are specified in the *Export Rectangle* boxes. The coordinate system defined by the *Export Rectangle* is the one used in the resulting file. When the dialog is opened, the boxes are automatically displayed with *Page Rectangle* and *Export Rectangle* coordinates that reflect the current page and map coordinate systems. If you change any of the values, you can reset them to the original settings using the *Default* button.

Coordinates used in the file can be offset and scaled relative to the default values by entering the new values into the boxes. In this way, you can scale the drawing to fit into the bounding rectangle at the same time you export it.

Scaling information can be specified in three ways:

- You can click the *Application* option button in the *Scaling source* group. This option automatically determines the page coordinates for the corners of the map, and determines the coincident map coordinates. This scaling information is then loaded into the appropriate boxes.
- Click the *Saved* option in the *Scaling source* group to reload previously saved scaling values. When the *Save scaling info* check box is active, the current *Page Rectangle* and *Export Rectangle* coordinates are saved. The *Saved* option loads those coordinates the next time you use the export option.
- You can type any values into the *Page Rectangle* and *Export Rectangle* boxes.

Spatial Formats

Spatial file formats are only concerned with the location of objects in space (i.e., their coordinates) and not with their properties such as line or fill style, marker symbol used, text labels, etc.

Boundary files such as Atlas Boundary [.BNA] files and Golden Software Boundary [.GSB] files are spatial formats. Some other formats, such as AutoCAD DXF files [.DXF], can be exported as a spatial format although the file type is not strictly a spatial format. Check the *Use ONLY spatial information* option to make these formats into spatial formats.

Suggestions on Selecting Export Formats

Refer to *Appendix C, File Formats* for more details on file types.

Here are a few tips on exporting and saving files.

- **Surfer** files [.SRF] save the entire contents of the plot window. [.SRF] files are only readable by **Surfer**, so they cannot be used in other programs. Surfer is backward compatible, which means **Surfer 8** can read **Surfer 5, 6, or 7** [.SRF] files. However, **Surfer** is not forward compatible, meaning **Surfer** files [.SRF] created in **Surfer 8** cannot be used in **Surfer 7, 6, 5**, or DOS.
- If you want the map coordinates in an ASCII format file, you can use the Atlas Boundary [.BNA] format or the Golden Software Blanking [.BLN] format.
- Within the Windows environment, Windows Picture (Enhanced) [.EMF] files are quite useful because they reproduce the map faithfully in the accepting application. This format is the same as copying a map to the clipboard and pasting it in the picture or metafile format.
- If you are creating a web site, you might need to create a [.GIF] or [.JPG] file. These formats are complete snapshots of a map.
- If you are exporting your map to the Macintosh environment, you might consider [.TIF] or [.PCT].
- If you are exporting to a CAD package, you might use the AutoCAD DXF option. [.DXF] files can be exported in map coordinates for overlaying on your CAD drawings.

Chapter 21

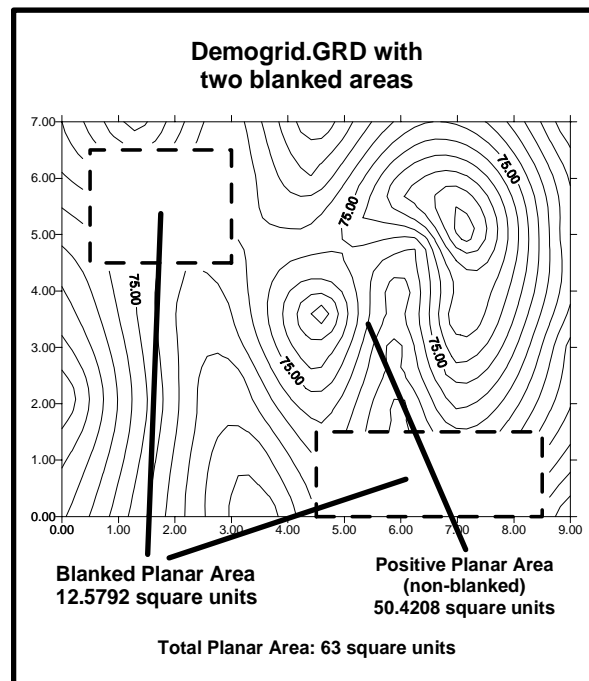
Creating, Selecting, and Editing Objects

Introduction to Creating, Selecting, and Editing Objects

Surfer plot documents contain a collection of individual graphic objects. These graphic objects include maps and drawing objects such as text blocks, lines, rectangles, etc. This chapter focuses on the creation of drawing objects, and the selection and editing of all objects (including maps). Refer to the individual map chapters for more information on creating each map type.

Creating Objects

The **Draw** menu contains several objects that can be added to the plot window. You can add text, polygons, polylines, symbols, rectangles, rounded rectangles, and ellipses through the **Draw** menu, or through the draw toolbar. Drawing object default fill properties, line properties, text properties, and symbol properties can be specified through **File | Preferences | Drawing**. Refer to the *Chapter 23, Preferences* for information on default draw properties.




Text can be added at any position in a plot window.

Text

You can use the **Text** command to create text in a document. The typeface, size, style, alignment, and color can be set for text blocks. Default text properties are set through the **File | Preferences | Drawing** command.

To create text:

1. Click the  button or choose **Draw | Text**.
2. When the cursor has the letter "T" next to it, click where you want text to appear in the document.
3. Enter text into the **Text** dialog.
4. Set the text properties in the **Text** dialog.
5. Click OK to return to the plot window.
6. Press the ESC key or click another button on the toolbar to end draw mode.

Using Math Text Instructions to Format Text

Math text instructions control the appearance of text in the plot window. Math text instructions change properties such as typeface, size, color, weight, and style (bold, italic, strikethrough, and underlined) within a single text block. You can also include Greek characters, superscript and subscript characters, italic or bold text, the current date or time, and different colored text in a single text block. Math text instructions are useful for placing mathematical equations in a plot window, or creating custom axis titles using mixed Greek and Roman characters.

The following is an example of how math text instructions can be used to modify a text block. The text block " $\delta^{234}\text{U}$ " is created by adding the text using the math text instructions as indicated by the following example.

$\delta^{234}\text{U}$ `{\f"Symbol" d}\up50 {\fs75 234}\dn50 U`


Each line in a text block begins with the properties specified in the **Text** dialog. Math text instructions cause variations in properties and position of text within the line. These variations can extend from the point of the instruction to the end of a line, or can be isolated to a few select characters within the line.

Math text syntax and instructions are discussed in detail in *Appendix B, Math Text Instructions*.

Polygon

You can use the **Polygon** command to draw an irregularly shaped area. Polygons must have at least three vertices (points).

To draw a polygon:

1. Choose the **Draw | Polygon** command or click the  button to begin drawing a polygon.
2. The pointer changes to a cross hair pointer to indicate digitize mode. Move the pointer over the location for the start of the polygon and click the left mouse button.
3. Move the pointer to the next position along the line and click again.
4. Continue this procedure until you click the final point, and then press the ENTER key.

The first and last points are automatically connected and the new polygon is drawn. To end drawing mode, click on another toolbar button or press the ESC key on your keyboard.


Drawing Tips

- Click points on the page to draw a polygon, or click and hold the left mouse button and drag the pointer to draw a continuous stream of points.
- Click the right mouse button to remove the last drawn point. You can right click repeatedly to remove several points.
- If the CTRL key is pressed while clicking points, the points are constrained to 45-degree angles.
- Double-click the left mouse button or press the ENTER key to close the polygon.
- To cancel drawing a polygon, press the ESC key before closing the polygon.
- After creating the polygon, you can edit the shape by using **Edit | Reshape**.
- Edit polygon line and fill styles by double-clicking on the polygon in the plot window or in the **Object Manager**.
- Line and fill default properties are set through **File | Preferences | Drawing**.
- If you prefer to use the tool only one time, check the *Single-use tool mode* box in **File | Preferences | General**. When *Single-use tool mode* is active, you can double-click on the tool to use it multiple times.

PolyLine

You can use the **PolyLine** command to draw a polyline. A polyline is a collection of one or more connected line segments.

To draw a polyline:

1. Choose the **Draw | PolyLine** command or click the  button to begin drawing a polyline.
2. The pointer changes to a cross hair pointer to indicate digitize mode. Move the pointer over the location for the start of the polyline and click the left mouse button.
3. Move the pointer to the next position along the line and click again.
4. Continue this procedure until you click the final point, and then press the ENTER key.
5. To end drawing mode, click another toolbar button or press the ESC key on your keyboard.


Drawing Tips

- Click the endpoints of the line to draw a straight line, or click several points to create an irregularly shaped line.
- Click the right mouse button to remove the last drawn point. You can right click repeatedly to remove several points.
- Click and hold the left mouse button to create a continuous stream of points.
- If the CTRL key is pressed while clicking points, the points are constrained to 45-degree angles.
- Double-click the left mouse button or press the ENTER key to end the line.
- To cancel drawing the line, press the ESC key before ending the line.
- After the polyline is created, you can edit the shape of the line using **Edit | Reshape**.
- Edit the line style and color by double-clicking on the line in the plot window or in the **Object Manager**.
- Set line property defaults through **File | Preferences | Drawing**.
- If you prefer to use the tool only one time, check the *Single-use tool mode* box in **File | Preferences | General**. When *Single-use tool mode* is active, you can double-click on the tool to use it multiple times.

Symbol

You can use the **Symbol** command to place symbols in the document. Symbols are markers that are used to indicate point positions. Any TrueType glyph can be used as a symbol, including several custom glyphs provided with **Surfer**.

To draw a symbol:

1. Click **Draw | Symbol** or click the  button.
2. The arrow pointer changes into a cross hair. Click on a location in the plot window to create the symbol.
3. Press the ESC key or click another toolbar button to end drawing mode.


Drawing Tips

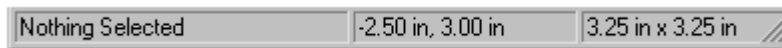
- Edit a symbol style, color, and size by double-clicking on the symbol in the plot window or in the **Object Manager**.
- Set symbol property defaults through **File | Preferences | Drawing**.
- If you prefer to use the tool only one time, check the *Single-use tool mode* box in **File | Preferences | General**. When *Single-use tool mode* is active, you can double-click on the tool to use it multiple times.

Rectangle

You can use the **Rectangle** command to create a rectangle or square in the document.

To draw a rectangle:

1. Select the **Draw | Rectangle** command or click the  button.
2. The arrow pointer changes into a cross hair to indicate drawing mode.
3. Press and hold the left mouse button at one corner of the rectangle.
4. Drag the mouse to the opposite corner of the rectangle. The size of the rectangle appears in the status bar as it is drawn. You can draw a rectangle out from the center rather than corner to corner by holding down the SHIFT key.
5. Release the left mouse button when the rectangle is the preferred size and shape.
6. Press the ESC key or click another toolbar button to end drawing mode.



The right-most pane of the status bar shows the size of the rectangle as it is drawn.

To draw a square:

Hold down the CTRL key while dragging the mouse to draw a square rather than a rectangle.


Drawing Tips

- Edit the rectangle or square properties by double-clicking on it in the plot window or in the **Object Manager**.
- Set line and fill default properties through **File | Preferences | Drawing**.

Rounded Rectangle

You can use the **Rounded Rect** command to create a rounded rectangle or rounded square in the document.

To draw a rectangle:

1. Select the **Draw | Rounded Rect** command or click the  button.
2. The arrow pointer changes into a cross hair to indicate drawing mode.
3. Press and hold the left mouse button at one corner of the rounded rectangle.
4. Drag the mouse to the opposite corner of the rounded rectangle. The size of the rounded rectangle appears in the status bar as it is drawn. You can draw a rounded rectangle out from the center rather than corner to corner by holding down the SHIFT key.
5. Release the left mouse button when the rounded rectangle is the preferred size and shape.
6. Press the ESC key or click another toolbar button to end drawing mode.

To draw a rounded square:

Hold down the CTRL key while dragging the mouse to draw a rounded square rather than a rounded rectangle.


Drawing Tips

- Edit the rounded rectangle or square properties by double-clicking on it in the plot window or in the **Object Manager**.
- Set line and fill default properties through **File | Preferences | Drawing**.
- If you prefer to use the tool only one time, check the *Single-use tool mode* box in **File | Preferences | General**. When *Single-use tool mode* is active, you can double-click on the tool to use it multiple times.

Ellipse

You can use the **Ellipse** command to create an ellipse or circle in the document.

To draw an ellipse:

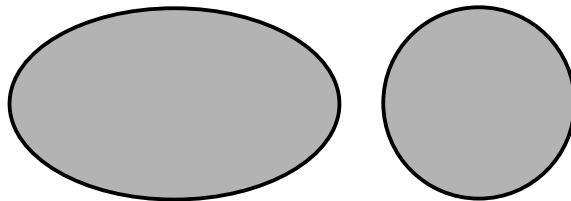
1. Select the **Draw | Ellipse** command or click the  button.
2. The arrow cursor changes into a cross hair to indicate drawing mode.
3. Press and hold the left mouse button at one corner of the bounding box of the ellipse.
4. Drag the mouse to the opposite corner of the ellipse. The size of the ellipse's bounding box appears in the status bar as it is drawn. You can draw an ellipse out from the center rather than corner to corner by holding down the SHIFT key.
5. Release the left mouse button when the ellipse is the preferred size and shape.
6. Click on another toolbar button or press the ESC key to end drawing mode.

To draw a circle:

Hold down the CTRL key while dragging the mouse to draw a circle rather than an ellipse.

Drawing Tips

- Edit the ellipse or circle properties by double-clicking on the object in the plot window or in the **Object Manager**.
- Set line and fill default properties through the **File | Preferences | Drawing** command.
- If you prefer to use the tool only one time, check the *Single-use tool mode* box in **File | Preferences | General**. When *Single-use tool mode* is active, you can double-click on the tool to use it multiple times.



*To draw an ellipse, click on the ellipse button, or choose **Draw | Ellipse**. To create a circle, hold down the CTRL key while using the **Ellipse** command.*

Selecting Objects


To perform any operation on an object, you must first select that object. Selected objects are indicated by a set of eight small green "handles." There are several methods to select objects using either the keyboard or mouse.

The status bar provides information about the selected object. For example, when you click on a contour map, the words "Map: Contours" appear on the left end of the status bar, indicating that a contour map is selected. When you click the bottom axis from the contour plot, the words "Map: Bottom Axis" appear in the status bar. (If you have changed object IDs, different titles appear.)

Under most circumstances, a command is applied only to selected objects and does not affect other objects. Some commands require a single object to be selected, while other commands can operate on multiple objects.

How to Select Objects

Several procedures are available to select objects:

- Objects may be selected using the **Object Manager**. To toggle the display of the **Object Manager**, use the **View | Object Manager** command or click the  button. Once displayed, click on the object in the **Object Manager** list that you wish to select. The object is then selected in the plot window. Note that some nested objects cannot be selected. An example of this is an object embedded within a composite object. You can hold down the CTRL key while clicking on objects to select multiple objects in the **Object Manager**, or hold down the SHIFT key to select a range of adjacent objects.
- To select a single object in the plot window, move the pointer over the object using the mouse or the arrow keys. When the pointer is over the desired object, click the left mouse button or press the SPACEBAR on the keyboard. Eight rectangular selection handles appear, indicating that the object is selected.
- If an overlapping object other than the one you want is selected, hold down the CTRL key and continue clicking with the mouse or SPACEBAR until the desired object is selected. Any objects that were previously selected become deselected.
- To select two or more objects in the plot window, hold down the SHIFT key while making your selections. This retains previously selected objects and includes the newly selected objects. You can hold down both the CTRL and SHIFT keys to select several overlapping

Chapter 21 - Creating, Selecting, and Editing Objects

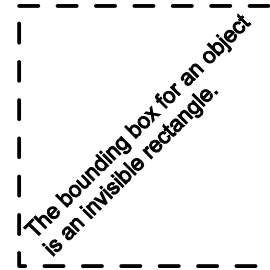
objects in the plot window. Clicking a selected object with the SHIFT key held down deselects that object.

- The block select mouse procedure allows you to select one or more objects contained in a user-defined rectangle. To "block select" an object, press and hold the left mouse button on an empty portion of the plot window, and then drag the mouse to form a rectangle around the group of objects you wish to select. Alternatively, you can use the arrow keys to position the pointer on an empty portion of the plot window, press and hold the SPACEBAR, and move the pointer with the arrow keys. If the *Rectangle must fully surround* check box is activated in **File | Preferences | Drawing**, then only objects fully surrounded by the selection rectangle are selected. If the check box is not activated, then all objects with any portion of their bounding boxes within the block select rectangle are selected. See *Chapter 23, Preferences* for more information on setting plot window preferences.
- The **Edit | Select All** command is used to select all the objects in the plot window. Pressing the CTRL+A keys performs the same command.
- The **Edit | Invert Selection** command selects all unselected objects and deselects all selected objects. This command is useful for selecting a large number of objects and leaving a few isolated objects unselected. Select the objects you do not want to select and use the **Invert Selection** command.
- The TAB key can be used to cycle through all objects, selecting each one at a time.

Bounding Boxes

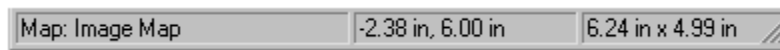
Object selection is based on an object's bounding box. The bounding box is defined as the smallest unrotated rectangle that fully surrounds the object. To select an object, the pointer must be within the object's bounding box.

Bounding boxes are invisible. When an object is selected, the selection handles indicate the limits of the bounding box. Also, when an object is moved or resized, the bounding box is indicated with a dashed line.



The bounding box for this rotated text block is indicated by the dashed line.

- When the pointer is in a window region occupied by two or more bounding boxes, the object drawn last is the selected object. You can change the order in which objects are drawn using the **Arrange | Order Objects** commands, giving you some control in object selection.
- When the bounding boxes for two or more objects occupy the same position, you can step through the list of objects by holding down the CTRL key and repeatedly clicking the mouse (or pressing the SPACEBAR) until you have selected the object you want. The status bar helps you to select the object you want by displaying information about each selected object.

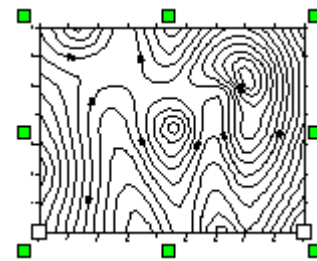


The left pane in the status bar identifies the selected object.

Selection Handles

Two types of selection handles can appear when an object is selected. The size of the selection handles can be controlled through **File | Preferences | Drawing**. See *Chapter 23, Preferences* for more information on preferences.

- Solid handles indicate that you can independently move or resize the selected object. The selection handles appear as eight solid green squares at the corners and on the sides of the bounding box for the selected object.
- Hollow handles show the current selection within a composite object, and indicate that the selected object cannot be moved or resized independently of the parent object. For example, if you select an axis on a contour map, solid handles appear surrounding the entire map, and hollow handles appear at the ends of the selected axis.



Eight solid handles appear around the selected composite object (the contour map), and hollow handles indicate that the bottom axis is the sub-selection.

Deselecting Objects

To deselect objects you can:

- Click in any empty space in the plot window, and the currently selected objects become deselected.
- Select any other object using the mouse or keyboard, and the currently selected objects become deselected.
- If you have several objects selected, you can deselect individual objects by holding down the SHIFT key and clicking the object to be deselected.
- All objects may be deselected using the **Edit | Deselect All** command.

Object Properties

The **Edit | Properties** command allows you to edit the properties of the selected object. The object properties are accessed:

- by selecting the **Edit | Properties** command when the object is selected,
- by double-clicking on the object in the plot window or in the **Object Manager**,
- or by right clicking on an object in the plot window or in the **Object Manager** and selecting the **Properties** command.

Modeless Property Dialog

The properties dialog can remain open while working in the plot window. The pages in the dialog change as the selection in the plot window changes. Before changing your selection, be sure to click the *Apply* button to save your changes (if any). The position of the property dialog is remembered and restored when the dialog is reopened. Also, the property dialog is retained when switching between plot windows.

Setting the Properties for a Single Object

To set the properties for a single object:

1. Select the object.
2. Open the properties using one of the methods described above.
3. Make the changes and click OK or *Apply*. Clicking the OK button applies the changes and dismisses the dialog. Clicking the *Apply* button applies the changes but leaves the dialog open for additional edits.

Setting the Properties of Multiple Selections

To set the properties for several selected objects at once:

1. Select all the objects to be changed.
2. Open the properties using one of the methods described above.
3. The **Properties** dialog only displays the properties that are shared by all the selected objects.
4. Make changes to the properties and click OK. The changes are made to all selected objects.

Tips on Properties of Multiple Selections


- If multiple objects are selected, the dialog contains the current values for the first selected object.
- If multiple selected objects (e.g. rectangle and contour map) are selected and they do not have any properties in common, the message *The current selection does not have any properties in common* appears. Change the selection to the object you wish to edit.
- If multiple maps are selected, only features common to all maps appear in the dialog, such as **View** and **Background**.
- Map-specific properties cannot be applied to multiple selected maps of a given type, for example, multiple contour maps. Map-specific properties, such as contour level intervals, have to be set on one map at a time.

Example 1

For example, if a solid red rectangle and a circle with no fill pattern are selected, the property changes can be made to the features these objects have in common. The rectangle and circle both have lines, so any changes to line properties are applied to both the rectangle and the circle. If the rectangle is the first selected object, the rectangle color fill can be changed from red to blue. Since the circle is not filled, the fill color is not applied to the circle.

Setting Fill, Line, Symbol, and Text Properties

Fill, line, symbol, and text properties can be specified for many objects. Line properties include color, thickness, and style. Line properties are applied to line objects such as contour lines, axes, map background borders, or to the borders around objects such as polygons, rectangles, rounded rectangles, and ellipses. Fill properties include fill patterns and colors, and can be applied to filled contour maps and enclosed objects such as rectangles, ellipses, and polygons. Symbol properties include symbol type, size, and color. Text properties include typeface, size, color, and style.

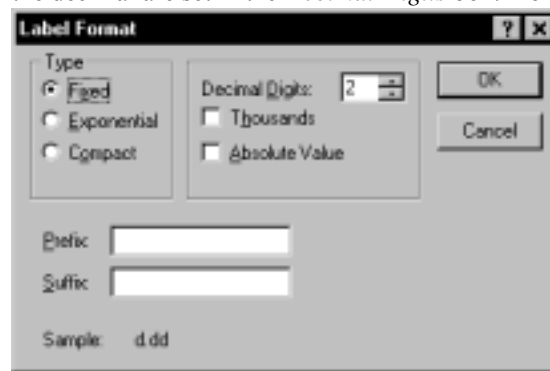
For specific information on fill, line, and symbol properties, use the online help with **Surfer**. For example, if you have a polyline's properties open and want information on the contents of the dialog, click the  button in the dialog.

Label Properties

The format of labels used by various objects may be modified using the **Label Format** dialog. The label type, length, prefix, and suffix may be set using this dialog. The dialog is typically displayed through a *Label Format* button in the object's properties dialog.

There are three number format types available in the *Type* group. The *Fixed* option displays numbers as dd.dd. The numbers to the right of the decimal are set in the *Decimal Digits* box. For example, if the numeric format is set to *Fixed* with three digits after the decimal point, the number 1998 displays as 1998.000.

The *Exponential* option displays numbers as d.ddE+dd where d is a single digit, dd can be one or more digits, and +dd is a sign and three digits. The numbers to the right of the decimal are set in the *Decimal Digits* box. For example, if the numeric format is set to *Exponential* with two *Decimal Digits*, then 1998 displays as 1.99E+003.




You can use the **Label Format** dialog to set the number of decimal places in labels.

The *Compact* option displays the labels in either fixed or exponential fashion, whichever requires fewer digits. Enter the number of digits to display in the *Significant Digits* box. For example, if the numeric format is set to *Compact* with two total digits, the year 1998 displays as 1.9E+003.

If the *Thousands* box is checked, a comma appears every three digits to the left of the decimal point. Check the *Absolute Value* check box to display the absolute value of the numbers. Negative numbers are displayed without negative signs.

You can add a text string before and after each label using the *Prefix* and *Suffix* boxes. For example, a "\$" could be used as a prefix, or a unit of measure "ppm" as a suffix.

Custom Line, Fill, and Color

You can click the *Custom* button at the bottom of the line palette, fill pattern palette, or color palette to create custom lines, fill patterns, or colors. After creating custom lines, fill patterns, or colors, you can add them to the palette or replace them in the palette. For details about the creating custom line, fill, and colors, click the  button in the dialog.

Setting Default Properties

You can specify text, fill, line, or symbol properties to use during the current session and subsequent sessions of **Surfer**. See *Chapter 23, Preferences* for more information.

Reshape

You can use **Edit | Reshape** to move, add, and delete vertices within a selected polyline or polygon. Polylines and polygons in broken apart metafiles can be also be edited with **Reshape**. After selecting **Reshape**, all vertices in the selected object are shown with hollow squares. You can click on a single vertex to select it. The selected vertex is indicated by a solid black square.

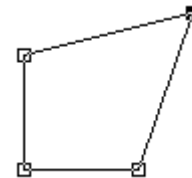
To edit a polyline or polygon:

1. Select the object.
2. Choose **Edit | Reshape**.
3. The arrow pointer turns into an arrowhead pointer and all the vertices appear as small hollow squares.
4. To move a vertex, click on the vertex with the mouse and drag it to a new location. To add a vertex, hold down the CTRL key and click a point on the polyline or polygon where you wish to add the vertex. To delete a vertex, select it and then press the DELETE key.

After reshaping the object, press the ENTER key to end edit mode, or press the ESC key to cancel the changes.



Click the Custom button at the bottom of the color palette to create new colors. You can also create new fill patterns and line styles by clicking the Custom button in the fill and line palettes.



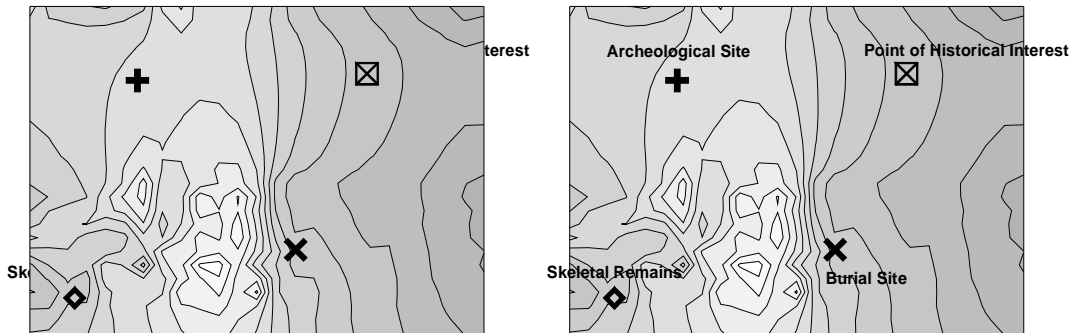
*When **Reshape** is selected, the vertices that make up the object are shown by small hollow squares. A selected vertex is shown as a solid black square.*

Arranging, Combining, Rotating, and Aligning Objects

Changing the Order in which Objects are Drawn

The **Arrange | Order Objects** command is used to change the order in which objects are drawn. Objects drawn last on the page might obscure objects drawn underneath. By selecting objects and choosing the **Arrange | Order Objects** command, you can change the order in which they are drawn.

- The **Move to Front** command moves all selected objects to the front of the drawing order. These objects are drawn last and appear on top of all other objects on the page.
- The **Move to Back** command moves all selected objects to the back of the drawing order. These objects are drawn first and appear beneath all other objects on the page.
- The **Move Forward** command moves the selected object forward one position in the drawing order.
- The **Move Backward** command moves the selected object backward one position in the drawing order.



*When objects are behind other objects, such as the text behind the filled contour maps in the example on the left, they are obscured by filled objects. You can use the **Arrange | Order Objects | Move to Front, Move to Back, Move Forward, and Move Backward** commands to change the order in which objects are drawn in the window. In the example on the right, the filled contour map is moved to the back, and the text can be seen.*

Changing the Order in which Maps are Drawn

When working with map overlays, you can select the overlay and choose the **Arrange | Order Overlay | Move to Front, Move to Back, Move Forward, and Move Backward** commands to change the order in which the overlays are drawn in the composite map.

Maps and objects can also be reordered in the **Object Manager**. To reposition an object in the **Object Manager**, click and hold the left mouse button on the object's name and then drag the object to a new position in the list.

Combining Objects

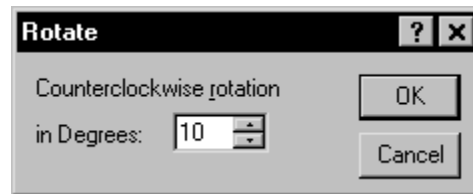
You can combine a group of objects into a single entity by using the **Arrange | Combine** command. This is helpful if you have several objects that you want to group together so you can move or resize them simultaneously. For example, you might create a title block consisting of a text block and a rectangle surrounding the text. If you combine the text block and rectangle, you can easily move or resize the composite object.

Objects in a composite cannot be edited. To edit an object from a composite you must first break apart the composite, and then you can make changes to the objects.


The **Arrange | Break Apart** command is used to separate objects previously combined into a composite object.

Rotating Objects

The **Arrange | Rotate** command opens the **Rotate** dialog. Positive numbers rotate the object in a counterclockwise direction and negative numbers rotate the object in a clockwise direction. If several objects are selected, they are rotated about the center of the bounding box. Individual objects are rotated about the center of the object.



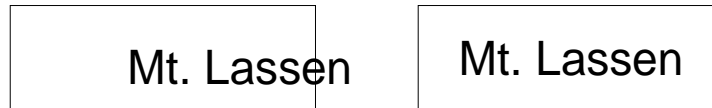
*The **Rotate** dialog allows you to specify the amount of rotation to apply to the selected object.*

The **Arrange | Free Rotate** command or the  button rotates selected objects around the center of the bounding box by dragging the mouse. Individual objects are rotated about the center of the object. The relative rotation is displayed on the status bar. When you choose this command, the cursor changes to indicate rotate mode. Press and hold the left mouse button inside the bounding box containing the selected object(s). Drag the mouse to rotate the bounding box. When the desired rotation is obtained, release the mouse button. Angles displayed in the status bar are in decimal degrees.

To cancel the current rotation, press ESC. To return objects to their original rotations, check the *Identity (Clear Transforms)* check box in **Arrange | Transform**. See *Advanced Transformations* for more information on **Transform**.

The **Arrange | Align Objects** command is used to align selected objects relative to the bounding box surrounding the objects. The objects can be aligned in both the vertical or horizontal directions. For example, you might create a title block consisting of a text block and a rectangle surrounding the text. If you select both the text block and the rectangle, you can choose the **Arrange | Align Objects | Center** command and then choose **Arrange | Align Objects | Middle**.

*The **Align Objects** command can be used to center a title relative to a map.*



When you want to align maps on the page, you can use the **Map | Stack Maps** command. The **Stack Maps** command aligns maps based on their coordinate systems as opposed to their bounding boxes. See *Chapter 17, Positioning and Overlaying Maps* for more information on stacking maps.

Object IDs

An object ID is an identifier or name that you can assign to any object in **Surfer**. When you select an object, the ID for the object is displayed in the status bar, the **Object Manager**, and in the title of the properties dialog.

When you create an object, a default object ID is assigned. For example, if you create a post map and then select the map, the status bar reports "Map: Post," telling you that a post map is selected. Consider the case where you combine two post maps into a single composite map, and want to select one of the post maps. If you assign object IDs to the two post maps, you can easily identify the two maps.

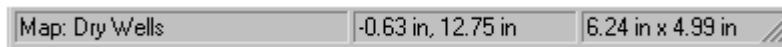
Assigning an ID to a Map

To assign an object ID to a map:

1. Create any type of map.
2. Select the map. The selected map is indicated in the status bar, with information such as "Map: Post" as well as in the **Object Manager** as "Post" and in the map properties title bar as **Map: Post Properties**.
3. Choose the **Edit | Object ID** command and the **Object ID** dialog is displayed. Enter any name into the dialog. The name can contain spaces if desired.
4. Click OK in the **Object ID** box and the status bar, **Object Manager**, and properties dialog titles are updated to display the Object ID. For example, if you entered a name such as "Dry Wells" for a post map, the status bar would report "Map: Dry Wells."



*You can name the selected object in the **Object ID** dialog.*



This is the status bar showing the selected Dry Wells object.

Deleting Objects

When an object is selected, you can delete it by pressing the DELETE key on the keyboard, or by selecting the **Edit | Delete** or **Edit | Cut** commands. If you wish to delete an overlay in a composite map, the overlay must be broken out of the composite map before deleting it. See page 492 for more information about breaking apart overlays.

Moving and Resizing Objects with the Mouse

When an object is selected, eight selection handles appear at the limits of the bounding box for the selected object or objects. When an object is selected, you can move it or resize it using the mouse.

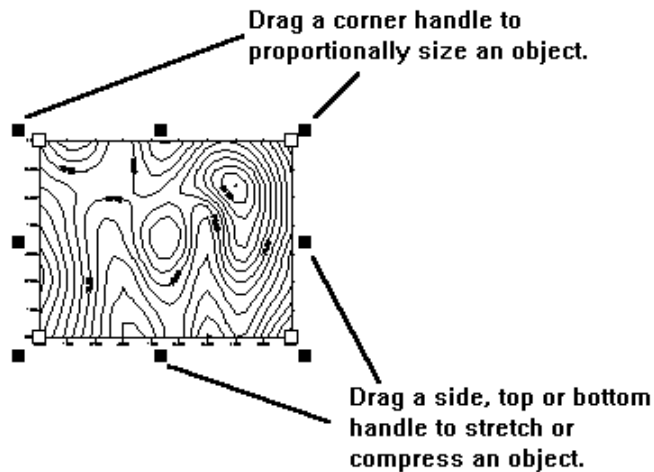
To move an object:

1. Select the object to be moved and selection handles appear surrounding the bounding box for the object. When you select more than one object, selection handles appear around the bounding box for the group of objects.
2. Place the mouse pointer inside the bounding box, then press and hold the left mouse button.
3. Drag the pointer to the position where you want the object to be redrawn. As you drag the object, a dotted rectangle indicates the limits of the object's bounding box. You can use the arrow keys on the keyboard to move the object in one-pixel increments if necessary.
4. When the bounding box is in the desired position, release the mouse button and the object is drawn in the new position.

To resize an object:

1. Select the object to be resized and selection handles appear surrounding the object.
2. To resize an object proportionally, drag one of the corner handles inward or outward. To resize non-proportionally, drag one of the corners while holding down the ALT key. To stretch or compress the object in one dimension, drag one of the side, top, or bottom handles inward or outward. A dotted rectangle indicates the size of the object as you are resizing.
3. Release the mouse button and the object is redrawn.

NOTE: If you graphically resize any text or symbol objects or maps using the mouse, you should be aware that the resizing information is not reflected back to the dialog for that object. For example, if you click a text block and resize it with the mouse, the original point size is still reported in the **Text Properties** dialog for that text block. If you stretch a map, the **Scale** dialog for that map still reports the original scaling information. Use **Arrange | Transform** to return to the original size. See *Advanced Transformations* for more information on **Transform**.



Using the Keyboard to Simulate Mouse Operations

You can use the keyboard to move the pointer within the plot window, to select and move objects, and double-click objects.

- The ARROW keys move the pointer within the plot window.
- Pressing the SPACEBAR is equivalent to clicking the left mouse button.
- "Double-clicking" the SPACEBAR is the same as double-clicking the mouse.
- Holding down the SPACEBAR and using the ARROW keys allows you to drag or resize selected objects.

Advanced Transformations

When you select and drag an object, or resize it via the sizing handles, you are transforming the object. The **Arrange | Align Objects**, **Arrange | Rotate**, and **Arrange | Free Rotate** commands also transform the selected object. The accumulated effects of all transformations applied to an object are recorded as a set of page transformation parameters.

The **Arrange | Transform** command allows complete control over the page transformation parameters for the selected objects. The page transformation parameters can be used to position, size, rotate, and shear a selected object. The **Arrange | Transform** command also allows you to reset the transformation for an object (i.e., remove all transformations). This is an advanced command and should not be used unless you are comfortable with coordinate systems and geometric transformations.

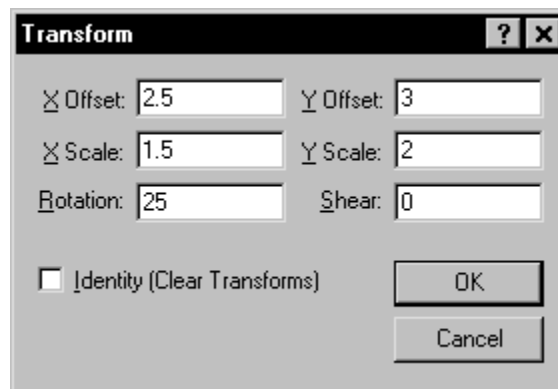
The **Arrange | Transform** command can be applied to any graphic object, including maps, text, polyline, and polygon objects.

Transform Dialog

The **Arrange | Transform** command activates the **Transform** dialog. This dialog allows you to read, set, and reset the six page transformation parameters associated with the selected object. These six parameters allow for four distinct transformations: scale, shear, rotation, and offset. The coordinate origin for these transformations is the center of the page. Thus, in the untransformed state, the center of an object coincides with the center of the printed page.

The page transformation parameters are as follows.

- The *X Offset* specifies the horizontal translation (in page units). A positive *X Offset* shifts the object to the right; a negative *X Offset* shifts the object to the left.
- The *Y Offset* specifies the vertical translation (in page units). A positive *Y Offset* shifts the object up; a negative *Y Offset* shifts the object down.
- The *X Scale* parameter modifies the width of the object. A value of 2.0



*The **Transform** dialog is used to remove scaling, rotation, and offsets.*

doubles the width, a value of 0.5 cuts the width in half, while a value of 1.0 leaves the width unchanged.

- The *Y Scale* parameter modifies the height of the object. A value of 2.0 doubles the height, a value of 0.5 cuts the height in half, while a value of 1.0 leaves the height unchanged.
- The *Rotation* specifies a counterclockwise rotation about the origin. The rotation angle is given in degrees.
- The *Shear* changes the X coordinate as the Y coordinate changes. For example, a shear of 1.0 shifts the X coordinates one inch to the right for every inch along the Y axis.
- When you check the *Identity (Clear Transforms)* check box, the transformation is reset back to the identity; that is, the selected object is untransformed.

The order in which the transformations are applied makes a difference (these transformations are not commutative). The four transformations are applied in the following order: scale, shear, rotation, and translation.

Transform References

Foley, James D. and Andries van Dam (1990), *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, MA, 1174 pp.


Hearn, D. and M.P. Baker (1997), *Computer Graphics C*, 2nd edition, Prentice Hall, Upper Saddle River, NJ, 652 pp., ISBN 0-13-530924-7.


Using the Object Manager

The **Object Manager** contains a hierarchical list of the objects in a **Surfer** plot window. The objects can be selected, arranged, and edited in both the **Object Manager** and through the menu commands. Changes made in the **Object Manager** are reflected in the plot document, and vice versa.

Opening and Closing the Object Manager

The **Object Manager** is opened and closed from the **View** |

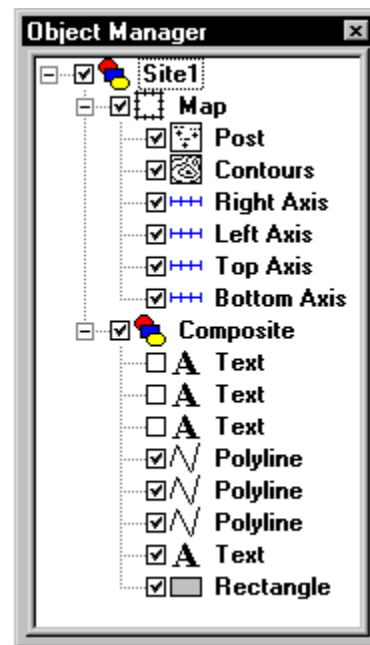
Object Manager command or from the  button.

Alternatively, you can click on the close window button  in the title bar of a floating **Object Manager** to close the window or click the **Edit** | **Object Manager** command to close a docked **Object Manager**.

Changing the Object Manager Location

The **Object Manager** can be docked on the edge of the **Surfer** window or floated as a pop-up window. To change the position of a docked **Object Manager**, click on the raised double ridge "grip" bar on the left or top of the **Object Manager** and drag it to a new location. A solid black rectangle indicates if the new location is docked. A thick light gray rectangle indicates that the new location is floating.

Double-click on the title bar or the raised double ridge to toggle between floating and docked modes. You can drag the sides of a floating **Object Manager** to change the size of the window.



The Object Manager displays all objects in the plot window.

Object Visibility

Each item in the list consists of an icon indicating the type of object and a text label for the object. All objects also have a check box that indicates if the object is visible. To change the visible status of an object click on the check box ☒ to the left of the object icon. A check in the box indicates it is visible; an empty box indicates that the object is not visible. Invisible objects do not appear in the plot window and do not appear on printed output. Note that if a surface is made invisible, the overlays also become invisible.

Object Manager Tree

If an object contains sub-objects a \oplus or \ominus is located to the left of the object name. Click on the plus or minus icon to expand or collapse the list. For example, a *Map* object contains a map overlay (i.e. *Contours*) plus four axes. To expand the tree, click on the \oplus control, or select the item and press the plus key (+) on the numeric keypad, or press the right arrow key on your keyboard. To collapse a branch of the tree, click on the \ominus control, or select the item and press the minus key (-) on the numeric keypad, or press the left arrow key. You can also double-click on a branch to expand or collapse the branch. The expansion state is retained in the **Surfer** file [.SRF]. Check *Expand new Object Manager items* in **File | Preferences | General** to force the **Object Manager** to expand all newly added objects in the tree control.

Selecting Objects

To select an item on the tree, click on the item or press the arrow keys, and the object text is highlighted. The selection handles in the plot change to indicate the selected item. To select multiple objects at the same level in the tree, hold down the CTRL key and click on each object. To select multiple contiguous objects at the same level in the tree, select the first object, and then hold down the SHIFT key and click on the last object. If you select an object in the plot window, its name is selected in the **Object Manager** as well.

When you select a nested object, the main object is also selected. For example, if you select the *Contours* overlay object, the *Map* object is also selected. Only one nested object can be selected at a time. For example, it is not possible to select two axes at once. It is also not possible to select any nested objects within a composite object. Break apart the composite object first to select and edit individual objects.

Opening Object Properties

To display the properties dialog for an object, you can double-click on most objects. Alternatively, you can right-click on that object and select **Properties**.

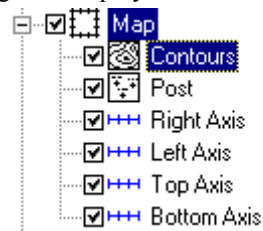
Editing Object IDs

To edit the text ID associated with an object, select the object, and then click again on the selected item (two slow clicks). You must allow enough time between the two clicks so it is not interpreted as a double click. Enter the new name into the box.

Arranging Objects

To change the display order of the objects with the mouse, select an object and drag it to a new position in the list above or below an object at the same level in the tree. The pointer changes to a black right arrow if the object can be moved to the pointer location or a black circle with a diagonal line if the object cannot be moved to the indicated location. For example, the *Post* map can be moved anywhere within the *Map* object, but not into a *Composite* object or another map.

To change the display order of the objects with the keyboard, select an object, hold down the



If a sub-object is selected (i.e. Contours), the main object (i.e. Map) is also selected.

CTRL key, and press the up and down arrow keys to move one step at a time. Hold down the SHIFT key and press the up and down arrow keys to move to the top and bottom of the same branch. These actions are analogous to the **Arrange | Order Objects** and **Arrange | Order Overlay** commands, which include the **Move to Front**, **Move to Back**, **Move Forward**, and **Move Backward** options. You can also access these menu commands by right clicking on the object in the **Object Manager**.

Deleting Objects

To delete an object, select the object and press the DELETE key. If the object is part of a higher-level object, then the entire higher-level object is deleted. For example, if you select *Contours* from the *Map* object, the entire *Map* object is deleted, including axes and other associated maps. To delete a *Contour* map without deleting the entire *Map* object, select *Contours*, choose the **Map | Break Apart Overlay** command, select the contour *Map* object, and then press the DELETE key.

Chapter 22

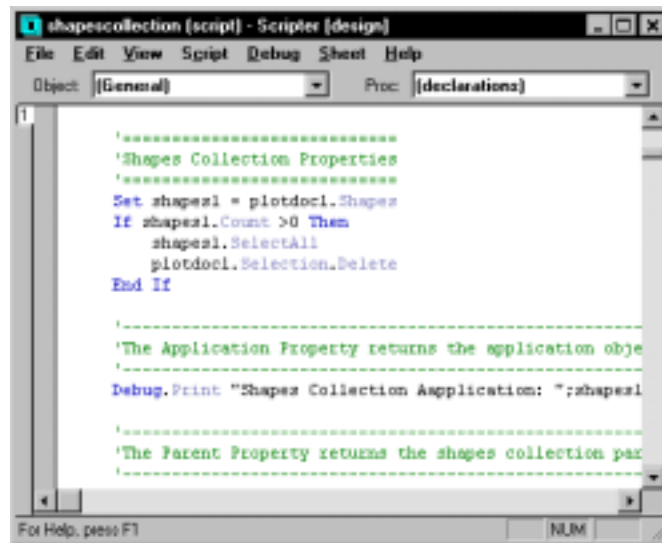
Automating Surfer

Introduction to Automation

Surfer operations can be controlled automatically by scripts. A script is a text file containing a sequence of instructions carried out by a script interpreter program. You can do practically everything with a script that you can do manually with the mouse or from the keyboard. Scripts are useful for automating repetitive tasks, consolidating a complicated sequence of steps, or acting as a "front end" to help novice users access **Surfer's** capabilities without having to become familiar with **Surfer**. In addition, scripts can integrate the features of several programs. For example, you could, with a single script, open a data file in Microsoft Excel, transfer the data to **Surfer**, create a map, and copy the map to Microsoft Word.

You can write scripts using the Golden Software **Scripter**[™] program that comes packaged with **Surfer**. Scripts are written

in a programming language that is compatible with Visual BASIC for Applications. **Scripter** provides a full-featured environment for writing, running, and debugging scripts. If you are already familiar with another programming environment, you may want to write scripts using an application other than **Scripter**. Because **Surfer** exposes its services through Automation, you can use any programming tool that can access Automation objects. Such tools include Visual BASIC, Windows Scripting Host, and many of the Microsoft Office applications, among others.



*This is a script in the main window of the Golden Software **Scripter** program.*

As an example, suppose you are monitoring groundwater levels in an area. Each week you collect XYZ data, create a contour map from this data, print the map, and save it in a **Surfer** file [.SRF]. The *Examples* section at the end of this chapter lists a script that automates these tasks. The sample script asks for the name of the XYZ data file, creates a grid file, produces a contour map, prints the map, and saves it in a file. A person using this "front-end" script would not have to know anything about **Surfer** in order to create a contour map from the raw water level data.

Surfer Automation Objects

The Surfer Object Model

Accessing Surfer

The means of accessing **Surfer** automation objects varies depending on the scripting tool and language being used. With the Golden Software **Scripter** program and other applications compatible with Visual BASIC, the `CreateObject` function creates a **Surfer Application** object:

```
Set x = CreateObject("Surfer.Application")
```

In this sample, a variable named "x" is assigned the value returned by the VB `CreateObject` function. The `CreateObject` function finds the name "Surfer.Application" in the system registry, automatically activates **Surfer**, and returns a reference to the **Surfer Application** object. For an introduction to the **Scripter** programming language, see *The Scripter BASIC Language* section of this chapter.

After creating an *Application* object, you can access other **Surfer** objects through the properties and methods (defined below) of the *Application* object. **Surfer** supplies nearly 60 different Automation objects. These objects are organized in a hierarchy, with the *Application* object at the root. The charts on the following pages show the arrangement of this hierarchy. This object hierarchy is known as an object model.

Methods and Properties

Objects have attributes associated with them. The *Grid* object, for example, has an attribute called `xMin` which represents the minimum X coordinate of the grid. An object's attributes, also known as its **properties**, may be examined and changed by scripts. Some properties may be examined, but may not be changed. These are called read-only properties.

In addition to having properties, many automation objects perform services. A *Grid* object, for example, performs the `SaveFile` service when a script instructs it to do so. The services that an

object performs are called its **methods**. Methods often require parameters to control how their services are performed. The *Grid* object's `SaveFile` method, for example, accepts two parameters: a file name and a file format code.

The way **Surfer** automation objects' properties and methods are accessed varies depending on the programming language you use. In the Golden Software **Scripter** language, properties and methods are accessed by typing the name of an object variable, followed by a period, and the name of the property or method:

```
Set x = CreateObject("Surfer.Application")
x.Visible = True
```

This sample creates an *Application* object and sets the *Application* object's `Visible` property to the special `True` value predefined by the **Scripter** language.

You can find all the properties and methods that are available for each of the automation objects, and a description of the objects, in the online help. To view the online help, select either the **Help | Automation Help** command in **Surfer**, or the **Help | Surfer Automation Help** command from **Scripter**. The **Scripter ActiveX Automation Members** dialog also shows you the available objects, properties, and methods. To view this dialog, select the **Debug | Browse** command in **Scripter**. See *The Object Browser* section of this chapter for information on this dialog.

Collections

Several non-visible objects, called **collection objects**, are used to organize the object hierarchy. Collection objects are containers for groups of other, related, objects. For example, the *Overlays* collection contains the overlays in a composite map. An *Overlays* collection object could contain a *BaseMap* object, *Wireframe* object, and a *PostMap* object if these three map objects were overlaid together.

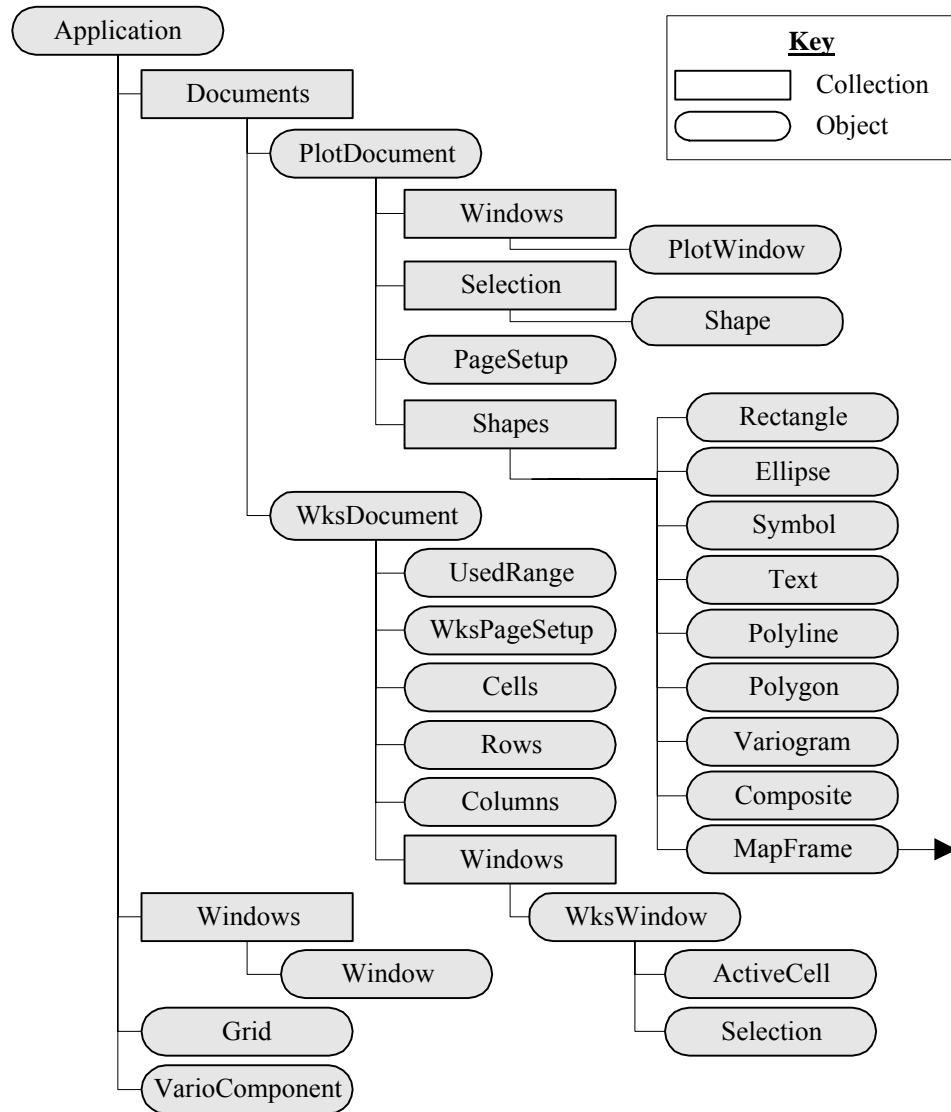
All collection objects, regardless of what the collection contains, can be processed using similar techniques. Thus, you retrieve an object from a collection of maps (an *Overlays* collection) the same way you retrieve an object from a collection of axes (an *Axes* collection). Collection objects also provide methods to create new objects.

Object Model Chart

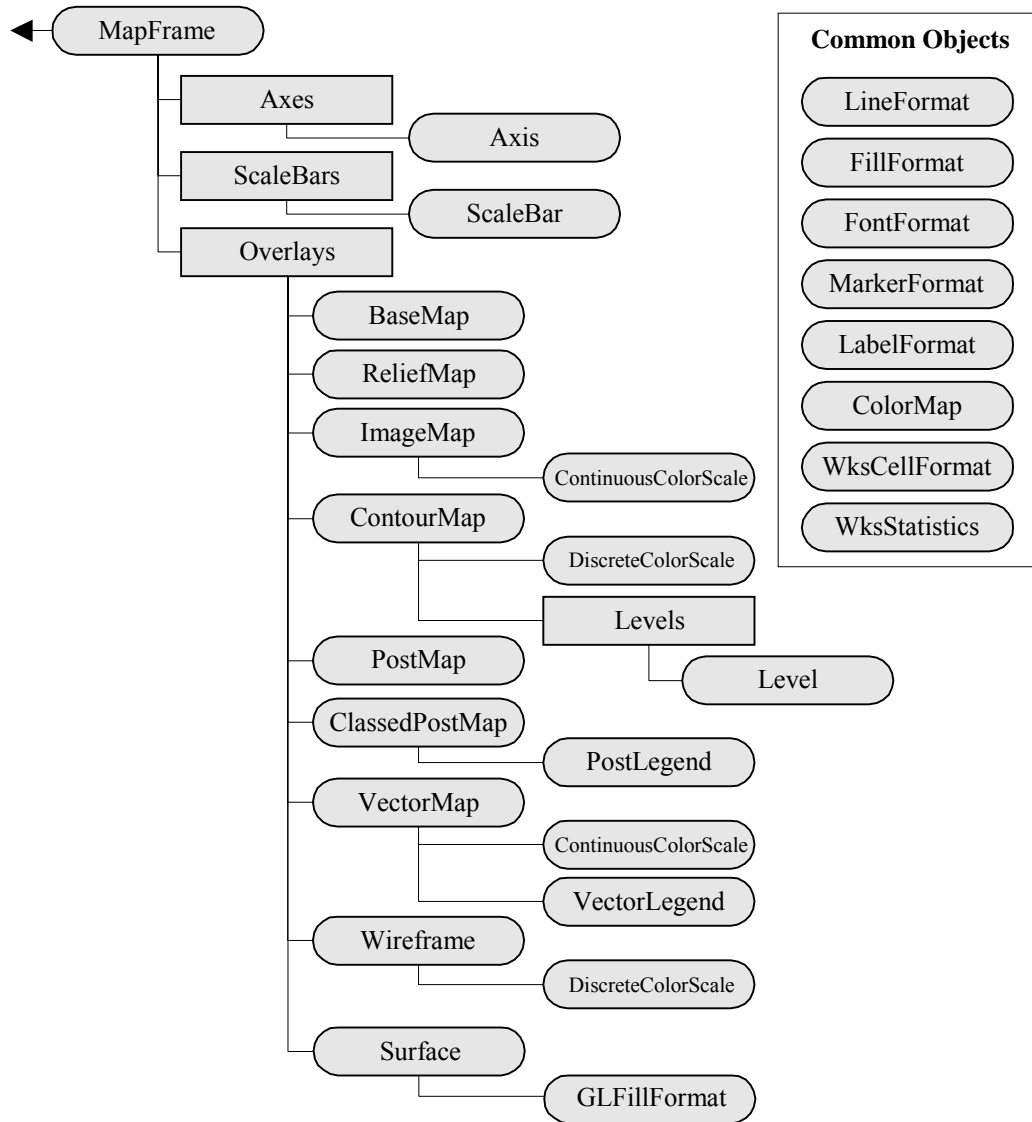
The object model chart shows you which objects provide access to other objects in the hierarchy. Although the *Application* object is at the top of the hierarchy, not all objects are directly accessible from the *Application* object. To access many objects you must traverse from the *Application* object through one or more layers of sub-objects. People often refer to "drilling" or "boring" through the object hierarchy to describe this traversal through several objects to obtain an object you want to use.

To "drill through" the object hierarchy you must know which properties and methods of an object provide access to the next level of objects. The next section, *Overview of Surfer Objects*, discusses the most commonly used objects and the properties and methods that provide access to other objects in the hierarchy.

Every object represents a specific part of **Surfer**. For example, the *Wireframe* object represents a wireframe, and the *Axis* object represents a map axis. Some objects do not represent a visible entity, but organize other objects into groups. The *Shapes* object, for example, provides the means to create new maps and new drawing primitives (rectangles, symbols, polygons, etc.), but the *Shapes* object itself is never visible.



This is the **Surfer** automation object model. To access a particular object you must traverse the object hierarchy using the methods and properties provided by the objects. The chart shows objects that provide access to other objects. Collection objects (shown as square-sided rectangles) are containers for groups of related objects. Non-container objects (shown as rounded rectangles) represent a specific part of **Surfer**.



This is the continuation of the **Surfer** automation object model chart. A **MapFrame** object is a type of **Shape** object under which all the **Surfer** map types are grouped. The "common objects" are contained by one or more automation objects. Most of the common objects are used to access a set of related properties in their parent object. Not every **Surfer** automation object is listed in these charts.

Overview of Surfer Objects

Learning to use the **Surfer** automation objects in a script may appear daunting at first. Most tasks, however, can be accomplished using just a few **Surfer** objects. Once you become familiar with these primary objects and learn how to "drill through" the object hierarchy, you will be able to access most of **Surfer's** features from your scripts.

The online help is the complete reference for all of the **Surfer** automation objects, their properties, and their methods. The object model chart should serve as your guide for navigating through the object hierarchy. Click **Help | Automation Help in Surfer**, or **Help | Surfer Automation Help in Scripter** to see the online reference.

This section shows how to access the **Surfer** automation objects using the **Scripter** BASIC language. Refer to the *Writing Scripts* section of this chapter for information about **Scripter** BASIC. If you are not familiar with computer programming, you may benefit from a programming tutorial. See the *Suggested Reading* section for recommendations.

Derived Objects

Several objects shown in the object chart share common features. For example, the *PlotDocument* and *WksDocument* objects each provide *SaveAs*, *Activate*, and *Close* methods. These common features are inherited from a predecessor *Document* object.

Derived objects inherit all the properties and methods of the predecessor object. In the online reference, the help topic for derived objects shows just the properties and methods that are unique to the object. The predecessor object is accessed through the *Derived from* link. Remember that all the properties and methods of a predecessor object are available as well. Derived objects and their predecessor objects include:

- The *Document* object is a predecessor of the *PlotDocument* and *WksDocument* objects.
- The *Window* object is a predecessor of the *PlotWindow*, *WksWindow*, and *GridWindow* objects.
- The *Shape* object is a predecessor of all objects that can be moved and resized. These include the *MapFrame*, *Variogram*, *Composite*, all of the basic drawing objects (*Rectangle*, *Ellipse*, *Symbol*, *Text*, *Polyline*, and *Polygon*), all of the map objects (*BaseMap*, *ImageMap*, *ReliefMap*, *ImageMap*, *ContourMap*, *PostMap*, *ClassedPostMap*, *VectorMap*, *Wireframe*, and *Surface*), and various map component objects (*Axis*, *ScaleBar*, *DiscreteColorScale*, *ContinuousColorScale*, *Legend*, *VectorLegend*, and *PostLegend*).
- The *Legend* object is a predecessor of the *VectorLegend* and *PostLegend* objects.

Chapter 22 - Automating Surfer

Using Collection Objects

Surfer groups most objects in collections. Although these collections contain different types of data, they can be processed using similar techniques.

Count Property

All collection objects have a read-only property named `Count` that gives the number of objects in the collection.

Item Method

Every collection also has a method called `Item` that retrieves one of the objects contained in the collection. The `Item` method accepts a single argument specifying either an index number or (for most collections) the name of the object to retrieve. An index number is a value between 1 and the value returned from the collection's `Count` property. The name used to identify items in a collection varies:

- In a *Documents* collection, the individual *Document* objects are identified by the file name or the caption, if the document has not been saved.
- In a *Windows* collection, individual *Window* objects are identified by the window caption.
- In collections containing objects derived from the *Shape* object (*Shapes*, *Selection*, *Axes*, *ScaleBars*, and *Overlays* collection objects), the individual *Shape* objects are identified by the object ID assigned to the object.
- In the *Levels* collection, individual *Level* objects may be retrieved by their index number only. The *Level* objects cannot be retrieved by name.

Shorthand syntax for retrieving items from a collection is to enclose the index or name of the desired object in parentheses immediately following the name of a collection object (similar to accessing an element in an array). Thus, the following two instructions have the same effect:

```
' Assume "docs" is a variable holding a reference to the Documents collection
docs.Item(1) ' retrieves the first object in the collection
docs(1)      ' shorthand, retrieves the first object in the collection
```

Objects contained in collections are typically derived from a base object. When you use the `Item` method to retrieve an object from a collection, the method returns a reference to the base object. To determine the actual type of the returned object, most base objects contain a `Type` method. For example, the *Windows* collection may contain *PlotWindow*, *WksWindow*, or *GridWindow* objects. When you retrieve a window from the *Windows* collection, a reference to a generic *Window* object is returned. Use the `window.Type` property to determine if the returned object is a *PlotWindow*, *WksWindow*, or *GridWindow* object.

Add Method

Many collections have one or more Add methods for creating new items and adding them to the collection. For example, to add a rectangle to the *Shapes* collection, you would use the `AddRectangle` method:

```
Shapes.AddRectangle 2, 2, 4, 4
```

Close and Delete Methods

The objects contained by collections are automatically removed when the contained object is deleted or closed. For example, call a *Document* object's `Close` method to close a document or call the *Shape* object's `Delete` method to remove any *Shape*-derived object.

Parent and Application Properties

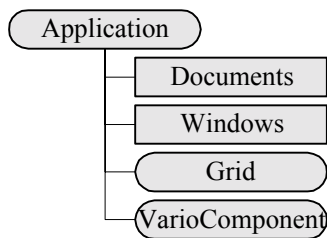
Every automation object provides a `Parent` and an `Application` property. The `Application` property returns a reference to the top-level *Application* object. This is a convenient way to access the *Application* object, particularly when passing automation objects as arguments to subroutine and function calls.

The `Parent` property returns a reference to the collection object that an object is contained in, or the controlling object. If an object is not contained by a collection object, the `Parent` property typically returns a reference to the *Application* object.

A Brief Introduction to the Major Surfer Objects

The following sections introduce some of the most important **Surfer** automation objects. This is by no means comprehensive. Please see the online help for a complete list of objects, properties, and methods.

The Application Object



All navigation through the **Surfer** object model begins with the *Application* object. When using VB or **Scripter** you can use the `CreateObject` function to obtain a reference to an *Application* object. The `CreateObject` function activates a new instance of **Surfer**, and returns a reference to the *Application* object to your script. If **Surfer** is already running, and you do not want to start a new instance of **Surfer**, use the VB `GetObject` function instead of `CreateObject`:

```
Dim srf As Object  
Set srf = GetObject(,"Surfer.Application")
```

The `GetObject` function obtains the *Application* object from the currently running instance of **Surfer**. If **Surfer** is not already running, the function will fail. Call the *Application* object's `Quit` method to close **Surfer** from a script.

When **Surfer** is started by a script, its main window is initially hidden. To make the **Surfer** window visible, you must set the *Application* object's `Visible` property to `True`:

```
Set srf = GetObject(,"Surfer.Application")  
srf.Visible = True
```

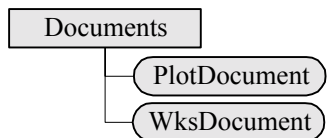
Most of the gridding related operations are methods of the *Application* object. Use the `GridData`, `GridFunction`, `GridMath`, `GridFilter`, `GridSplineSmooth`, `GridBlank`, `GridConvert`, `GridVolume`, `GridSlice`, `GridTransform`, `GridExtract`, and `GridCalculus` methods to create new grids and perform computations with grids.

Other methods and properties of the *Application* object let you move and resize the **Surfer** window, adjust the main window state (maximized, minimized, hidden), change the window caption, and modify preference settings.

The *Application* object provides two important collections that allow access to the next level of objects in the hierarchy. Use the Documents property to obtain a reference to the *Documents* collection object, and use the Windows property to obtain a reference to the *Windows* collection object. The *Documents* and *Windows* collection objects provide access to the other objects in the object hierarchy. You can access the currently active document and window objects with the *Application* object's *ActiveDocument* and *ActiveWindow* properties.

Set the *ScreenUpdating* property to *False* to disable screen updating and make your scripts execute more quickly. Turning off screen updating can greatly increase the performance of automation when you want to perform a number of actions that cause screen redraws.

The Documents Collection



The *Documents* collection provides the means to access plot and worksheet documents. Use the *Add* method to create new plot or worksheet documents. **Surfer** provides the pre-defined values *SrfDocPlot* and *SrfDocWks* for specifying the type of document to create:

```
Set srf = CreateObject("Surfer.Application")
srf.Documents.Add SrfDocPlot ' Create a new, blank plot document
srf.Documents.Add SrfDocWks  ' Create a new, blank worksheet document
```

The *Add* method returns a reference to the *Document* object that was created. New grid documents are not created with the *Documents* collection *Add* method. Create grid files by calling the *Application* object's grid-related objects.

Use the *Open* method to open existing plot, worksheet, and grid files. The *Open* method returns a reference to the opened document.

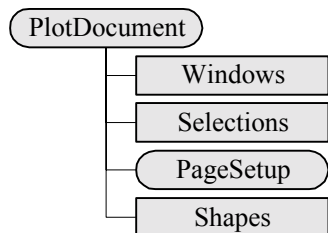
Chapter 22 - Automating Surfer

Use the *Item* method to retrieve a reference to an open plot or worksheet document object. When iterating through a collection of documents, the document *Type* and *Index* properties can be used to determine the type and position of the document within the collection.

The Windows Collection

The *Windows* collection provides access to all windows within **Surfer**. The *Windows* collection returned by a plot or worksheet document just lists the windows on that specific document. The *Windows* collection returned from the *Application* object lists all windows, no matter with which document they are associated. To create new *Window* objects, call a *Document* object's *NewWindow* method.

The PlotDocument Object

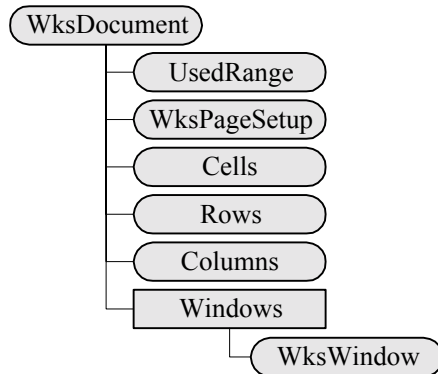


The *PlotDocument* provides access to all the drawing and map objects. Use methods and properties in the *PlotDocument* object to save **Surfer** files [.SRF], to import and export files, to print, and to change the default line, fill, font, and symbol for a plot.

Use the *Shapes*, *Selection*, and *Window* properties to access the collection objects having the same names. These collection objects provide access to the next layer of objects in the object hierarchy. Use the *Shapes* collection object to create new drawing and map objects.

The *Windows* collection contains all the windows open on the plot document. The *Selection* collection contains all the shapes currently selected.

The WksDocument Object



The *WksDocument* object represents a worksheet. Perform a mathematical transformation on worksheet cells, and merge files together with the *WksDocument* object. Navigate to the *Range* object using the Cells, Rows, Columns, and UsedRange properties. Get and set cell values, modify cell formats, sort, and calculate statistics using the *Range* object.

The PlotWindow, WksWindow, and GridWindow Objects

It is important to distinguish between windows and documents. *Document* objects represent a file on disk containing the actual data to be represented. Windows are used to view that data. There can be multiple windows viewing the same document at once.

The *Window* object encapsulates all the methods and properties associated with a **Surfer** window. Methods exist to open, close, move and size the window.

When a document is closed, all windows viewing that document are closed as well. The reverse relationship exists as well. When the last window viewing a document is closed, the document is closed.

Chapter 22 - Automating Surfer

The Shapes Collection

Within automation, graphic drawing and map objects are called Shapes. The *Shapes* collection object contains all of the drawing and map objects in a document. Methods exist to enumerate all the shapes in the collection, add objects to the *Selection* collection, and create new drawing and map shapes.

One of the most common uses of the *Shapes* collection is to create new shapes. There is a unique method to create each type of shape. For example, to create a contour map use the *AddContourMap* method, as in:

```
Sub Main
    Set srf = CreateObject("Surfer.Application")
    srf.Visible = True

    ' Create a new, empty plot document
    Set Doc = srf.Documents.Add SrfDocPlot

    ' Get the Shapes collection
    Set Shapes = Doc.Shapes

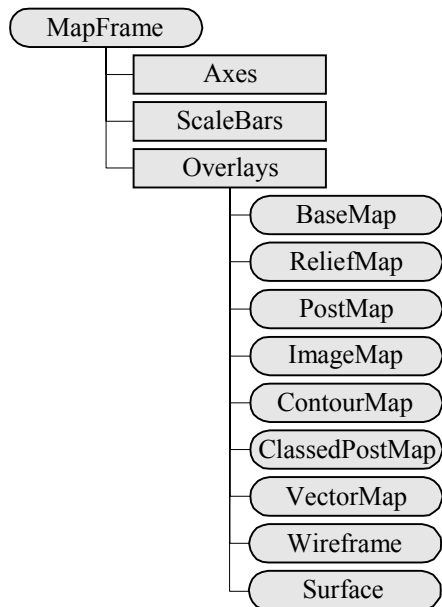
    ' Create the contour map
    Set MapFrame = Shapes.AddContourMap("Demogrid.grd")
End Sub
```

The Selection Collection

The *Selection* collection contains all currently selected shapes. To select a particular shape, set the shape's *Selected* property to True. When a *Shape* object is selected, it is added to the *Selection* collection. When it is deselected, it is removed from the *Selection* collection. Note that the shape is neither created nor destroyed by selection operations. It is merely added or removed from the *Selection* collection.

The *Selection* collection allows all selected shapes to be enumerated, and contains most methods that operate on multiple objects. This includes moving and sizing one or more shapes; cut, copy, and paste; and overlaying and stacking multiple maps.

The MapFrame Object



New maps are created using the `Add<map type>` methods of the *Shapes* collection (`AddBaseMap`, `AddReliefMap`, `AddContourMap`, and so forth). These methods return a reference to a *MapFrame* object - not, as you might expect, a reference to the object of the specific map type that was created. The *MapFrame* object contains the coordinate system, axes, and all properties common to the maps contained within.

Use the *Overlays* property of the *MapFrame* to obtain the *Overlays* collection object. The *Overlays* collection contains all the individual map objects associated with a map frame. When a new map frame is created, the *Overlays* collection contains just one map item. To obtain a reference to a newly created map, access the first item of the collection:

```
Sub Main
    Set srf = CreateObject("Surfer.Application")
    srf.Visible = True

    ' Create a blank plot
    Set plot = srf.Documents.Add srfDocPlot
```

Chapter 22 - Automating Surfer

```
' Add a new contour map. This returns a MapFrame object
Set mapframe = plot.Shapes.AddContourMap "demogrid.grd"

' Get a ContourMap object from the MapFrame
Set contourmap = mapframe.Overlays(1)
End Sub
```

Use the *MapFrame* object to set map limits, adjust the three-dimensional view, adjust the map scale, and change the size of the map. Access the *Axes* collection with the *MapFrame* object's *Axes* property. Use the *Axes* collection object to obtain the individual *Axis* objects associated with a map. Access the *ScaleBars* collection through the *MapFrame* object's *ScaleBars* property. Use the *ScaleBars* collection to add scale bars to a map and to access the *ScaleBar* objects associated with a map.

Surfer Version 6 Object Model

The **Surfer** version 6 automation model is not available in **Surfer** 8. Convert your scripts to the new model to make them work with **Surfer** 8.

Introducing Scripter

Golden Software **Scripter**™ is a program for developing and running scripts. A script is a text file containing a series of instructions carried out when the script is run. Instructions are written in a Visual BASIC-like programming language.

Scripter offers many features to help you write, edit, and debug scripts. Its features include language syntax coloring, a drop-down list of the procedures defined in the script, an object browser for examining procedures available in external objects, a visual dialog editor, break points, single-step execution (including options to *step over* and to *step out of* procedures), a watch window for displaying the values of script variables, and others.

To start the **Scripter** program, select it from the Windows Start menu. **Scripter** is installed in the same program group as **Surfer**. If **Scripter** is not present in the Windows Start menu, installation of **Scripter** may have been skipped when **Surfer** was installed. You can install **Scripter** from the **Surfer** CD-ROM. See *Chapter 1, Introducing Surfer*, for details on the installation process.

Scripter Windows

When **Scripter** is first started, you are presented with a text editor window containing the lines `Sub Main`, followed by a blank line, and then `End Sub`. This is the code editor window where you type script instructions and where the contents of script files are displayed.

The code window acts as a text editor, similar to the Windows Notepad program, with a few enhancements to facilitate script writing:

- After you press the ENTER key, tabs or spaces are automatically inserted at the beginning of the next line to maintain the same indentation level as the previous line.
- Key words and symbols of the BASIC language are displayed in different colors. You can use the **View | Colors** command to modify the colors used to display the various elements of the programming language.
- A light horizontal divider line is automatically drawn between sections of your script. The divider lines help you to locate the start of subroutine and function definitions.

Above the code editor window is a bar containing the *Object* and *Proc* (procedure) lists. Selecting items from these lists moves the various sections of your script file into view. The object and procedure lists are useful when your script file becomes large.

Above the object and procedure lists, you may see a blank window area with a tab on top that reads **Immediate**. If this window is not visible, select the **View | Always Split** command to make it appear. The immediate window is used to execute one-line instructions immediately. When you type an instruction into this window and press the ENTER key, **Scripter** carries out the instruction.

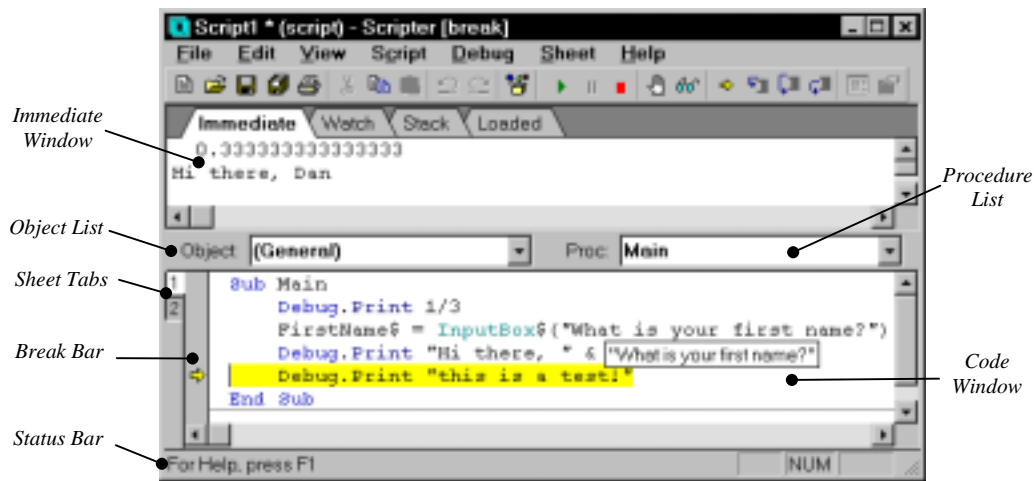
In addition to being a scratch area for evaluating language statements, the immediate window shows debugging information. The output from the `Debug.Print` statement and the value of variables selected with the **Debug | Quick Watch** command are printed in the immediate window. While a script program is running, **Watch**, **Stack**, and **Loaded** tabs are added at the top of the immediate window area. Click these tabs for information that may be useful for debugging. See the *Debugging Scripts* section of this chapter for more information on the immediate, watch, stack, and loaded windows.

Along the left edge of the code window are code sheet tabs. When you select either the **File | New** command or the **File | Open** command, **Scripter** creates a new code sheet and inserts a new sheet tab. Each tab corresponds to one of the code sheets. Clicking once on a tab makes that sheet the current sheet. Double-clicking a tab closes the sheet.

Chapter 22 - Automating Surfer

Between the sheet tabs and the code window is an area called the "break bar." When a script is paused, a yellow arrow in the break bar shows which line is next to execute. The break bar also shows which lines have break points. Set a break point by clicking on the break bar. A red dot appears in the break bar, and the adjacent line in the code window is highlighted. When a line marked as a break point is about to be executed, **Scripter** pauses program execution. To clear a break point, click on the red dot in the break bar. See the *Debugging Scripts* section for more information on break points.

A status bar along the bottom of the **Scripter** window shows information about the current state of the program. The **View | Status Bar** command hides or reveals the status bar. Before running a script, make sure that the status bar is visible because messages about typographical and syntax errors are displayed in the status bar.



This Scripter application window is shown while execution of a script is paused.

Working with Scripts

New Scripts

To create a new script, select the **File | New** command. A blank sheet is created. You can start typing script instructions into this sheet. If you edit more than one sheet at a time, click the sheet tabs to switch between them or select the **Sheet | 1**, **Sheet | 2**, etc. menu commands. You can edit up to nine code sheets at the same time.

New Modules

To create a custom Automation object, select **File | New Module**, and choose either **Object Module** or **Class Module** (choosing **Code Module** is the same as the **File | New** command).

Existing Scripts and Modules

To open an existing script, select the **File | Open** command. To open a script you opened recently, click its name at the bottom of the **File** menu. To open other modules used by your script, select the **Sheet | Open Uses** command.

Saving Scripts

Once a script is complete, you can save the script by using the **File | Save** or **File | Save As** menu commands. If a script has not been changed since the last save, the **Save** command is grayed out.

Closing Scripts

To close the active script, use **File | Close**, use **Sheet | Close**, or double-click the sheet tab of the sheet. Close all open scripts with **Sheet | Close All**.

Using Scripter Help

For information on **Scripter** program menu commands, select the **Help | Contents** command in **Scripter**. Press the F1 key for information about the **Scripter** windows or the active dialog. The **Help | Surfer Automation Help** command shows all **Surfer**-specific methods and properties.

The online help, shown when you select the **Help | BASIC Language Help** command (or press SHIFT+F1), explains all of the BASIC language statements and functions. Each help topic describes the purpose of a statement, and shows the syntax (the order of keywords and other symbols) to use when writing an instruction. The syntax examples in the online help use a shorthand method to explain the possible variations in usage:

Chapter 22 - Automating Surfer

<code>Sub, End, True</code>	Words with the initial letter capitalized indicate language-specific keywords.
<code><u>name</u></code>	Lower-case words with a dotted underline are placeholders for information you supply.
<code>[param]</code>	Items inside square brackets are optional. These may be omitted from the statement.
<code>{Until While}</code>	Items between curly braces, and separated by a vertical bar are lists of options. You must use one of the items in list.
<code>[Private Public]</code>	Items between square braces and separated by a vertical bar are lists of optional choices. You may use one of the items in the list, or none of them.
<code>...</code>	An ellipsis mark indicates that the preceding item in the syntax example is repeated. Repeated items are usually optional and separated by commas.
<code> ; , . ()</code>	Other symbols must be typed as shown in the syntax example, with the exception of the underscore "_" character, which is used to show that a sample line has been split.

Writing Scripts

To create a script, you must type the script text into the **Scripter** code window, or edit an existing script. When you want to create a new script that is specific for your circumstances, you will most likely start with an empty **Scripter** window and type the entire script. If you want to perform a routine task such as creating a grid file or a contour map, you can probably open an existing script file and edit the file to meet your specific needs. **Surfer** comes with several sample scripts that you can modify as desired.

Consider a script that creates a grid file from an XYZ data file, and then creates a contour map from the grid file:

```
Sub Main
    ' Create a programmable object to represent the Surfer program
    Set srf = CreateObject("Surfer.Application")

    srf.Visible = True           ' Make the Surfer window visible
    Set plot = srf.Documents.Add ' Add a new plot document
    DataFile$ = GetFilePath$()    ' Prompt for name of the data file
```

```
' Invoke the Surfer Grid Data command. Note that the "_" character
' indicates that the instruction continues on a second line
srf.GridData DataFile := DataFile$, _
    OutGrid := "out.grd", ShowReport := False

' Invoke the AddContour map method. The AddContourMap method is a
' member of the Shapes collection object. The Shapes collection
' object, in turn, is one of the properties of the plot document object.
plot.Shapes.AddContourMap "out.grd"
End Sub
```

When you execute the script, **Surfer** is automatically started and a plot window is displayed. The grid file is created (the progress of the operation is indicated on the **Surfer** status bar), and the contour map is drawn in the plot window. When the script execution is complete, you are returned to the **Scripter** window. The **Surfer** window remains open, although the **Scripter** window is the active window.

The Scripter BASIC Language

This section describes the major elements of the **Scripter** BASIC programming language, but it does not explain the concepts of writing computer programs. Many good books on the subject of programming with BASIC (**B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode) have been written. If you are not moderately familiar with writing computer programs, we suggest that you refer to one of the books listed in the *Suggested Reading* section.

Scripts are text files that contain a sequence of instructions to be carried out by the **Scripter** program. Each instruction specifies a task such as defining a variable or displaying a message on the screen. When the **Scripter** program processes the script, the instructions are carried out one at a time, from top to bottom.

Execution of a script begins with the first statement of the subroutine called `Main`. All scripts must therefore include the `Sub Main` and `End Sub` statements. Execution proceeds line-by-line until the end of the `Main` procedure, until an `End` statement is executed, or until an error occurs.

Chapter 22 - Automating Surfer

Program Statements

Statements are individual instructions to **Scripter** that carry out a specific operation. Statements are not case sensitive and are typically written one to a line. To enter two or more statements on the same line, separate the statements with colons. For example:

```
a = 5 : b = 5 * a
```

Scripter BASIC requires flow control statements (If, While, Do, etc.) and declaration statements (Dim, Public, Type, Sub, etc.) to be placed on a line by themselves.

Line Continuation

Long statements, particularly function calls requiring a large number of arguments, often will not fit on a single line. A long line may be split into two or more lines by adding a space followed by an underscore " _" at the end of each line that is continued:

```
Function ComputeSomething( filename As String, _  
value_array() As Double) As Double
```

Comment lines cannot be continued with an underscore; start each comment line with an apostrophe when writing multiple-line comments.

Comments

Writing comments in your scripts to explain how they work can save you time and frustration when you later need to modify the script. The apostrophe character (') signals the start of a comment. **Scripter** ignores all text following the apostrophe up to the end of the line. Comments can be placed on their own line, or they may be placed at the end of a line. For example:

```
srf.GridData "sonora.dat" 'This creates a grid file from sonora.dat
```

In addition, you can use the REM statement to add a remark in the script. However, REM statements can only be used at the beginning of a line.

Double Quotes and Text

Text strings must be enclosed in double quotes. File names, for example, must be surrounded by double quotes. If quotes are missing, the text may be mistaken for a variable name.

```
Debug.Print "This text string is printed in Scripter's immediate window"
```

Variables

A variable is a symbolic name for a value. A variable name starts with a letter and may contain digits. Variable names cannot be the same as a reserved word. Because the **Scripter** code window displays variable names in black and reserved words in color, you can see when you have selected a variable name that conflicts with a reserved word.

Variables may be one of several types. The type of a variable determines what kind of data it may contain. See the following table for the possible variable types. In addition to the built-in data types, the **Scripter** language supports user-defined compound data types, user-defined enumeration types, and user-defined objects (defined in object modules and class modules). See *User-Defined Types* and the *Code, Class, and Object Modules* section of this chapter for more information. Refer to the **Scripter** online language help for enumerations.

The type of a variable is declared in a `Dim` statement. The syntax of a `Dim` statement is:

`Dim varname As type`

where `varname` is the name of the variable being declared, and `type` is the variable's data type. Variables not declared in a `Dim` statement are a Variant type, unless the variable name ends with one of the type-definition characters. If a variable name ends with one of the special type-definition characters, listed in the next table, its type is recognized based on this character.

Type	Type-definition Character	Description of Type
Integer	%	A 16-bit integer value
PortInt (Portable Integer)	?	A 16- or 32-bit integer value
Long	&	A 32-bit integer value
Single	!	A 32-bit floating-point value
Double	#	A 64-bit floating-point value
Currency	@	A 64-bit fixed-point value
String	\$	A text string of any length
Byte	(none)	An 8-bit unsigned integer value
Boolean	(none)	A true or false value
Date	(none)	A 64-bit floating-point value
Object	(none)	A reference to an object
Variant	(none)	Capable of holding any type of value

Chapter 22 - Automating Surfer

Using the `Dim` statement to declare the variable type is optional. Variables can be used without first being declared in a `Dim` statement, but this practice is not recommended for any script longer than a few dozen lines. To enforce this policy, an `Option Explicit` statement should be placed at the top of long scripts. The `Option Explicit` statement makes it an error to use any variable without first declaring it. Using this option lets you find typographical errors in variable names before a script is run. Without this option, typographical errors in variable names are usually detected only when the script fails to produce the expected results.

Object Variables

Object variables contain references to objects. Creating the **Surfer Application** object is an example of declaring an object variable:

```
Dim SurferApp As Object
Set SurferApp = CreateObject("Surfer.Application")
```

In this example, a `Dim` statement declares that the variable named `SurferApp` will hold a reference to an object. The built-in `CreateObject` function returns a reference to a **Surfer Application** object, and the `Set` statement assigns this object reference to the `SurferApp` variable. Unlike variables of other types, which can be assigned new values simply with an equal sign (`=`), object variables must be assigned values with a `Set` statement.

Array Variables

Array variables store a list or table of values. A single variable name refers to the entire collection, and individual values are distinguished by their numeric indices (their "subscripts"). The maximum number of values that can be stored in an array must be defined in a `Dim` statement. The elements of an array are accessed by using the variable name followed by a left parenthesis, the index of an array element, and a right parenthesis.

```
Dim month(11) As String
month(0) = "January"
month(1) = "February"
...
month(11) = "December"
```

Array subscripts begin with zero, unless an `Option Base` statement is used at the start of a script. Notice that in the previous example an array whose maximum subscript value is 11 actually has room for twelve elements because the subscripts start with zero.

The `Dim` statement can reserve only a constant number of elements for an array. If the maximum number of elements cannot be known in advance, a *dynamic array* may be used. A dynamic array is an array whose number of elements can be changed while a script is running. The `Redim` statement changes the maximum number of values that can be stored in a dynamic array. Refer to **Help | BASIC Language Help in Scripter** for more information on `Dim` and `Redim`.

User-Defined Types

A collection of related variables can be grouped together under one name. The `TYPE` statement defines the elements of a user-defined type.

```
Type measurement
    julianday As Integer
    level As Double
End Type
```

The `TYPE` definitions must appear at the top of a script file, before any subroutines. The `TYPE...END TYPE` statement defines a new type; it does not create a variable of that type. Variables of the user-defined type must be declared in a `Dim` statement. The elements of a user-defined type variable are accessed by using the variable name followed by a period and the element name:

```
Dim m As measurement
m.julianday = 192
m.level = 12.3
Debug.Print m.julianday ' prints 192 in the Immediate window
Debug.Print m.level     ' prints 12.3 in the Immediate window
```

Global Variables

Variables declared in the body of a subroutine or function are available only within that procedure. If you want to share the same variable throughout a script file, then you can define it at the top of the file, before any subroutine definitions. Variables declared at the top of the file are available to all subroutines in the file; hence, they are called "global" variables.

The `Public` keyword may be substituted for the `Dim` keyword to allow a global variable to be used in other modules. See the *Code, Class, and Object Modules* section of this chapter for more information on modules.

Operators

Operators are symbols that direct a script to perform basic calculations, such as addition, exponentiation, string concatenation, number comparison, and others. The language supports several arithmetic, comparison, and logical operators. Select the **Help | BASIC Language Help** command in **Scripter** and search for *Operators* to see a complete list.

Flow Control

When you run a script, execution starts with the `Sub Main` statement and continues line-by-line until the `End Sub` statement at the end of the main procedure or until an `End` statement is encountered. Several flow control statements allow you to change this line-by-line progression according to conditions encountered by your script. The **Scripter** BASIC language includes a variety of looping and branching statements that is typical for modern programming languages. The flow control statements include the following (see the online BASIC language help for details on the syntax of these statements):

IF...END IF

IF...END IF executes a statement only if a condition is true. The alternate IF...ELSE...END IF form executes one statement if a condition is true and a different statement if the condition is false.

SELECT CASE...END SELECT

SELECT CASE...END SELECT branches to one of several statements. This compares the value of an expression to several test values and executes the statements associated with the test value that matches the expression value.

DO...LOOP

DO...LOOP is the basic looping statement. This statement loops either while a condition is true or until a condition becomes true and tests the condition either at the top of the loop or at the bottom of the loop.

This and all other loop structures may be stopped before the test condition has been met by placing an `Exit` statement in the body of the loop.

WHILE...WEND

WHILE...WEND loops while a condition is true and tests the condition at the top of the loop.

FOR...NEXT

FOR...NEXT loops a number of times and increments (or decrements) an index variable each time through the loop.

FOR EACH...NEXT

FOR EACH...NEXT iterates through all the elements in a collection object. Several **Surfer** automation objects are collection objects. The For...Each statement is a convenient way to process each element in a collection. For example, the following code fragment closes all the documents contained in the **Surfer Documents** collection object:

```
' Assume that several documents are already open and that "SurferApp"
' is the name of a variable which refers to the Surfer Application object
Dim documents, doc As Object
Set documents = SurferApp.Documents
For Each doc In documents
    doc.Close
Next
```

Optional Arguments and Named Arguments

Many procedures, especially the methods provided by **Surfer** automation objects, accept a large number of arguments. Some of the arguments are required. Every required argument must be supplied or the script will fail to run. Some arguments are optional. Optional arguments may be omitted and the procedure assumes a default value for the missing arguments.

For example, the *PlotDocument* object's PrintOut method accepts up to seven arguments, all of which are optional:

```
plot.PrintOut(Method, SelectionOnly, NumCopies, Collate, _
    xOverlap, yOverlap, Scale )
```

Since the arguments are optional, you can skip all or some of them when calling the procedure. To print three copies at fifty-percent scale, for example, you would supply just the NumCopies and Scale argument values. These arguments must be listed in the correct position, separated by commas, as shown below:

```
Set srf = CreateObject("Surfer.Application")
Set plot = srf.Documents.Add
plot.PrintOut , , 3, , , , 50
```

Chapter 22 - Automating Surfer

Although only two of the seven argument values are supplied in this example, the appropriate number of commas must be used to mark the positions of the missing arguments. Since inserting the right number of commas can be troublesome, you can supply the arguments by name rather than by position. Named arguments are specified by the argument name followed by a colon and an equal sign (`:=`), followed by the argument value:

```
Set srf = CreateObject("Surfer.Application")
Set plot = srf.Documents.Add
plot.PrintOut Scale := 50, NumCopies := 3
```

Named arguments may be listed in any order without regard to the order they appear in the procedure's definition.

Subroutines and Functions

Writing a long or complicated script may be easier to manage if you divide the script into smaller pieces called procedures. A procedure is a separate sequence of instructions that you can call from multiple places within your script. The BASIC language provides many pre-defined procedures for performing frequently needed tasks, and, in fact, the methods provided by the **Surfer** automation objects are themselves procedures.

When you call a procedure, the instructions in the procedure are executed. When the procedure finishes its task, it returns control to the instruction that called the procedure. The **Scripter** BASIC language distinguishes two types of procedures: functions and subroutines. Functions return a value, whereas subroutines do not.

Subroutines and functions may accept one or more values, called arguments. Arguments are passed to a procedure by listing them after the procedure name. If there is more than one argument, the arguments must be separated by commas. For example:

```
x = Cos(0)           ' Returns the cosine of the argument (returns 1)
a$ = Left("AgAuPb",2) ' Returns the left-most characters (returns "Ag")
Wait 5               ' Waits for 5 seconds
```

`Cos`, `Left`, and `Wait` are procedures built-in to the BASIC language. `Cos` and `Left` are functions which return values to the caller. `Wait` is a subroutine, and, by definition, it does not return a value. The `Wait` subroutine waits for the number of seconds specified by its argument (5 seconds in this example) before returning control to the calling location.

The arguments passed to a function must be enclosed in parentheses if the function's return value is used. If the function's return value is not used, the arguments may be listed without enclosing them in parentheses. Arguments passed to a subroutine are never enclosed in parentheses.

Writing Subroutines

To define subroutines within a script, use the `Sub` statement. Subroutine and function definitions cannot be nested within other procedures. That is, the `Sub` statement must appear after the `End Sub` statement of any preceding subroutine definitions. The syntax for a subroutine definition is:

```
Sub name ( arguments )  
    statements  
End Sub
```

where name represents the name you want to give to the subroutine, arguments represents a list of arguments names and types, and statements represents the instructions that comprise the body of the subroutine. There is no limit to the number of instructions you can include between the `Sub` and the `End Sub` lines. Consider the definition of a `Main` procedure and another subroutine:

```
Sub Main  
    MultipleBeep 25    ' call the MultipleBeep subroutine  
End Sub  
  
Sub MultipleBeep (count As Integer)  
    For i = 1 To count  
        Beep  
        Wait 0.5    ' Wait one-half second between beeps  
    Next  
End Sub
```

Each time the `MultipleBeep` procedure is called, the instructions between its `Sub` and `End Sub` statements are executed.

If the subroutine accepts arguments, the arguments are defined following the subroutine name using a format similar to the `Dim` statement. The argument definition list must be enclosed in parentheses, and argument definitions are separated by commas if there is more than one. When a subroutine is called, the variables listed in the argument list are automatically assigned the values passed in from the calling procedure.

Chapter 22 - Automating Surfer

Writing Functions

Functions are defined using the `Function` statement much the same as subroutines are defined with the `Sub` statement. Like subroutines, function definitions cannot be nested within other procedures. Unlike subroutines, functions can return a value to the calling procedure. The syntax of a function definition is:

```
Function name ( arguments ) As type  
    statements  
End Function
```

where name is the function name you want to use, arguments is a list of arguments names and types, type is the type of the value returned by the function, and statements are the instructions in the body of the function. To return a value from a function, assign a value to a variable with the same name as the function itself. For example:

```
Function hypotenuse(a As Double, b As Double) As Double  
    c = a * a + b * b    ' The built-in Sqr function computes the square root  
    hypotenuse = Sqr(c) ' Set the function's return value  
End Function
```

You define the list of arguments accepted by a function the same way as you define the arguments accepted by subroutines.

Built-in Functions and Procedures

Numerous useful functions and subroutines are built into the **Scripter** BASIC language. These routines can help you perform some of the most commonly required programming tasks. Functions for processing strings, performing mathematical calculations, error handling, working with disk files, and many others are available.

If you are not already familiar with the Visual BASIC for Applications programming language, it will be worth your time to review the list of available routines. This list is found by selecting **Help | BASIC Language Help in Scripter**.

Using Surfer Objects

To access **Surfer** commands from your script you must create a **Surfer Application** object as described in the *Surfer Automation Objects* section of this chapter. Every object has properties and methods associated with it. Properties are values describing the state of an object. Methods are actions an object can perform. Access properties and methods by typing the name of an object variable, followed by a period, followed by the property or method name.

You can use object properties as you would use variables: assign values to properties, branch based on the value of a property, or use the value of a property in calculations. You call an object's methods as you would call subroutines and functions. Use the return values from methods the same as you would use return values from functions.

When you "drill through" the object hierarchy, you can store references to intermediate objects in variables, or you can string together long sequences of object references. For example, you can set the default font for a plot document in a single line:

```
' Assume "srf" is a variable holding a reference to the Application object
srf.Documents.Item(1).DefaultFont.Bold = True
```

Alternatively, you can store each intermediate object in variables as you traverse the object hierarchy:

```
' Assume "srf" is a variable holding a reference to the Application object
Set docs = srf.Documents
Set plot = docs.Item(1)
Set font = plot.DefaultFont
font.Bold = True
```

The second form - storing intermediate objects - is more efficient if you are performing several actions with the same object. A third alternative is to use the `With...End With` statement:

```
' Assume "srf" is a variable holding a reference to the Application object
With srf.Documents.Item(1).DefaultFont
    .Bold = True
    .Size = 12
    .Color = srf.ColorBananaYellow
End With
```

Getting User Input

The **Scripter** language provides several predefined functions for prompting the script user for information. The `GetFilePath` function displays a standard Windows file open dialog. This allows the script user to select a file. The name of the selected file is returned by the function. The `InputBox` function allows the user to enter an arbitrary line of text. This function returns the line of text entered.

```
UserText$ = InputBox("Enter something here:") ' show prompt
Debug.Print UserText$                       ' show line in Immediate window
```



*This is a simple example of the InputBox function.
The text entered in the box is returned to the script.*

In addition to these simple input routines, **Scripter** supports user-defined dialogs. You can design your own dialogs and process their input in any manner you choose (see *Creating Dialog Boxes*).

Code, Class, and Object Modules

If you create very large scripts, or frequently reuse the same subroutines in several different scripts, you may want to split your script code into more than one file. Each script file is called a module.

A script can call subroutines and functions defined in other modules. In addition to procedures, global variables, type definitions, and enumeration definitions may be shared among modules. Just as procedures make long scripts easier to manage and debug, modules make large script projects easier to manage.

Module Types

The **File | New Module** command in **Scripter** adds new code sheets to the workspace. Each sheet is stored in a separate file. When routines in one code sheet are used by other sheets, the code sheets are called modules. **Scripter** supports three types of modules:

- Code modules are used for stand-alone scripts and for storing libraries of useful procedures that can be called from other modules. The scripts described in this chapter are code modules, which contain a `Main` subroutine. Code modules without a `Main` subroutine cannot be run, but the routines contained in them can be shared by other scripts. Code modules are stored in files with a `[.BAS]` extension.
- Class modules are used to define objects that you can use in other modules. A class module defines the properties and methods that the object supports. Other modules access the object's properties and methods using the same syntax that is used to access **Surfer** automation objects. Unlike **Surfer** objects, new instances of the object defined in a class module are created using the `NEW` keyword. Class modules are stored in files with a `[.CLS]` extension.
- Object modules are identical to class modules, except that when a script uses the object defined in an object module, one instance of the object is automatically created. Additional instances of an object defined in an object module can be created with the `NEW` keyword. Object modules are stored in files with an `[.OBM]` extension.

The '#Uses Line

Before using the procedures and objects defined in another module, a script must indicate the name of the file containing the procedure or object definitions. You must place a `#Uses` statement at the beginning of a script, before any procedure definitions, to instruct **Scripter** to load all modules used by the script. For example:

```
'#Uses "c:\utils.bas"
'#Uses "test.cls"
Sub Main
    ' use the procedures and object defined in UTILS.BAS and TEST.CLS
End Sub
```

Scripter does not permit cyclic `#Uses` statements. That is, if module A uses module B, module B cannot use procedures from module A.

Private and Public Definitions

By default, all subroutines, functions, and user-defined types (including enumeration definitions) may be accessed from other modules. To prevent other modules from accessing procedures or user-defined types precede the definition with the `Private` keyword:

```
Private Sub MyBeep
    Beep : Beep
End Sub
```

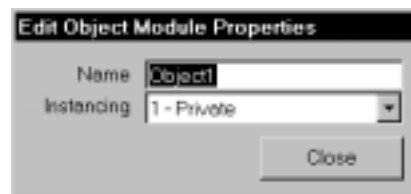
In contrast to procedures, the global variables defined at the top of one module are not available to other modules unless they are declared using a `Public` statement. When used for variable declarations, the `Public` statement has the same syntax as the `Dim` statement:

```
Public universal_data As String
```

The names of all definitions, even private ones, are visible in other modules. To avoid errors due to name conflicts you must avoid using the same procedure, type, and variable names in more than one module. A common technique for avoiding name conflicts is to append a prefix to the names of global variables and procedures. For example, if you write a module of text-processing functions, you might prefix each function name with `txt` (e.g., `txtFunction1`).

Module Properties

To set the name by which other modules refer to an object defined in a class or object module, select the **Edit | Properties** command. The **Edit Class Module Properties** dialog appears. Type the name that you want other scripts to use when referring to the object defined in the module. The instancing options control how other applications access the object defined in the module, but these options are not relevant to scripts executed within **Scripter**. Code modules do not have module properties.



*When working with an object module or class module, open the **Edit Object Module Properties** or **Edit Class Module Properties** dialog with **Edit | Properties**.*

When an object module is used in a script, one instance of the object defined in the module is automatically created. The name of the object that is automatically created is the name specified in the **Edit Object Module Properties** dialog.

Defining Object Properties and Methods

Class and object modules define the properties and methods of objects. To define the methods for an object, simply define public subroutines and functions. All the public procedures in a class or object module are methods that users of the object can call.

The properties of an object typically correspond to private global variables defined in the module. To allow users of the object to access the variable values, you provide "property get" and "property set" procedures. Use the `Property Get` statement to define a function that returns the value of a property. Use the `Property Let` statement (or the `Property Set` statement if the property is an object reference) to define a subroutine that changes the value of a property.

Two special subroutines are called when an object is first created and just before it is finally destroyed. In a class module, these subroutines are called "Class_Initialize" and "Class_Terminate." In an object module, these subroutines are called "Object_Initialize" and "Object_Terminate." These subroutines do not take any arguments.

Example: Defining an Object in a Class Module

The following class module demonstrates how to define an object. The sample defines a property named `Radius` and a method named `Draw`.

```
' Declare a private global variable for storing the property called "Radius"
Dim cirRadius As Double

' Define the initialization subroutine
Private Sub Class_Initialize
    cirRadius = 99
End Sub

' Define the termination subroutine
Private Sub Class_Terminate
End Sub

' Define the "property get" function to retrieve the Radius property
Property Get Radius() As Double
    Radius = cirRadius
End Property
```

Chapter 22 - Automating Surfer

```
' Define the "property let" procedure to change the Radius value
Property Let Radius(val As Double)
    cirRadius = val
End Property

Sub Draw
    ' Method performs some action here
End Sub
```


Scripts that use the class module would access the object as follows (assuming the module is stored in the file CIRCLE.CLS and that the object name entered in the **Edit Object Module Properties** dialog is MyCircleObject)


```
'#Uses "circle.cls"
Sub Main
    Dim x As New MyCircleObject
    x.Radius = 7          ' sets the value of the Radius property
    x.Draw               ' calls the Draw method
    Debug.Print x.Radius ' prints 7 in the immediate window
End Sub
```

Creating Dialog Boxes

Scripter contains a dialog box editor that you can use to design customized dialog boxes. Select **Edit | UserDialog Editor** to visually design a dialog box. You can control the size and placement of the components of the dialog box, as well as customize the text included in the dialog box.

To add a component to a dialog, first select from the palette of components at the left side of the **UserDialog Editor**. After clicking a palette button, drag the mouse pointer diagonally in the dialog design area where you want to place the component. As you design the dialog, you can edit the properties of components you have placed in the dialog. To edit the properties of a component, double-click the item, click the right mouse button on the item, or select the

component and click the  button. See the table on page 530 for a list of dialog components (also known as *dialog controls*) and their properties. Every dialog must include an OKButton, a CancelButton, or both.

When you have finished designing the dialog, click the  button. The code for the dialog is inserted into the script. To edit the dialog template after it has been inserted into the script, first move the cursor in the code window to any line between the `Begin Dialog` statement and the `End Dialog` statement. Next, select the **Edit | UserDialog Editor** command. The previously saved state of the dialog is shown in the dialog editor. When you save the dialog again, the previous dialog template is replaced with your changes.













To show your custom dialog in a script, first use the `Dim` statement to declare a variable as the `UserDialog` type, and then call the `Dialog` function to display the dialog (see the example below). The `Dialog` function takes a user dialog variable as its argument and returns a number indicating which button was clicked to end the dialog. The `Dialog` function returns -1 if the OK button was clicked, 0 if the *Cancel* button was clicked, or an integer greater than zero if a push button was clicked (1 for the first push button listed in the dialog template, 2 for the second push button in the dialog template, and so forth).

If the return value is not needed, the `Dialog` instruction may be called as a subroutine rather than as a function (in this case, do not enclose the dialog variable in parentheses). If the `Dialog` instruction is called as a subroutine, however, the script ends with a run-time error if a cancel button is clicked.

To define more than one custom dialog in a script, you must place each dialog template in its own subroutine or function. If you try to define more than one custom dialog in the same procedure, **Scripter** shows an error indicating that the `UserDialog` type has already been defined.

The values contained by dialog controls are accessed the same the way the fields of user-defined variable types are accessed. Type the dialog variable name, followed by a period, followed by the field name of the dialog component. Option button values cannot be accessed directly, but are accessed via the field name of their associated option group. The value of an `OptionGroup` is the number of the selected option button (the first option button in the group is 0, the second option button is 1, and so forth). You can initialize the values contained by dialog controls prior to showing the dialog, and retrieve the values entered in the dialog after it has been invoked.

Chapter 22 - Automating Surfer

Item	Description	Properties
 GroupBox	A rectangle used to group related controls.	Field, Caption
 Text	A text label, requires no response from the user	Field, Caption
 TextBox	An edit box used to enter and edit text	Field, Type (single line, multiple line, password)
 CheckBox	A box which is checked and unchecked as the user clicks on it, a three-state check box has a disabled (grayed) state	Field, Caption, Type (2 state, 3 state, 3 state auto check)
 OptionButton	A round button for choosing from a set of options, only one of a group of option buttons may be checked	Field, Caption, Option Group
 ListBox	A window that contains a list of items that can be selected by the user	Field, Array of Items, Type (list, sorted list)
 DropListBox	A list that is visible when opened by the user, the text may be editable or not and the list may be sorted or not	Field, Array of Items, Type (list, text, sorted list, sorted text)
 ComboBox	A text box with an attached list box, the list may be sorted or not	Field, Array of Items, Type (text, sorted)
 Picture	Displays a bitmap in the dialog	Field, Caption, Type (from file, from clipboard)
 PushButton	A push button	Field, Caption
 OKButton	Push button with the OK caption	Field
 CancelButton	Push button with Cancel caption	Field
Dialog	Definition of dialog box	Dialog Function, Caption, Centered

Properties:

- **Dialog Function:** The dialog function is the name of a special function that is called when various events happen in a dialog. Define a dialog function to control the behavior of a dialog and to retrieve its input.
- **Field Name:** The field name is the name used to refer to a component.
- **Caption:** The caption is the text displayed within a component.
- **Quoted:** When not quoted, the caption property gives the name of variable that contains the text to display for the caption. When quoted, the caption property is the literal text to display.

- **Array of Items:** The array of items is the name of a string array variable that contains the strings to display in a list. The array variable must be initialized before the dialog is invoked.
- **Type:** The type is the behavior of some components varies depending on which option is selected. Refer to the online help for descriptions of the available component types.
- **Option Group:** For option buttons, the field name is used to refer to a group of option buttons. Only one option button within a group may be checked.
- **Comment:** A comment is the text to insert in the dialog definition block.

Example

The following function demonstrates how to define, display, and extract the values entered in a user dialog.

```
Function MyInputDialog As String
    ' Define the dialog template. This
    ' definition is inserted by the
    ' UserDialog editor.
    Begin Dialog UserDialog 250,112,"Caption"
        TextBox 10,14,230,28,.Text1
        CheckBox 20,49,160,14,"Check _
            Box",.Check1
        OKButton 20,77,90,21
        CancelButton 130,77,90,21
    End Dialog

    ' Declare a dialog variable
    Dim dlgvar As UserDialog

    ' Initialize the dialog controls
    dlgvar.Text1 = "This is the initial text to display"
    dlgvar.Check1 = True ' start with check box checked

    ' Display the dialog and wait for the OK or Cancel button to be pressed
    result = Dialog(dlgvar)

    ' Extract the information entered into the dialog
    If result = -1 Then ' check to see if OK button was pressed
        MyInputDialog = dlgvar.Text1
        If dlgvar.Check1 Then Debug.Print "The Check Box was Checked!!"
    End If
End Function
```



This script creates the dialog displayed above.

To perform processing while a user dialog is active, define a special dialog function. The dialog function is called when various dialog events occur. To define a dialog function:

1. While designing the dialog, double-click in a blank portion of the dialog design area to activate the **Edit UserDialog Properties** dialog.
2. Enter a name for the Dialog Function property of the dialog. This property gives the name of a function that is called when dialog events occur.
3. When you save the dialog, **Scripter** asks if it should create a skeleton dialog function. Click the *Yes* button, and **Scripter** inserts the basic instructions for a dialog function into your script.

Refer to the DialogFunc help topic in the **Scripter** BASIC language help for more information about how to process dialog events in a dialog function.

Type Library References

Many application programs expose their services with objects. These objects may be used in your scripts, just as the **Surfer** automation objects are used. Before you can use another application's objects, you must add a reference the application's type library. A type library is a file that describes the objects, properties, and methods provided by an application. The **Surfer** type library is automatically referenced.

To add a type library reference to the current script module:

1. Select the **Edit | References** command to open the **References** dialog.
2. Click on the check box next to the type library whose objects you want to use in your script.
3. Click on the up and down arrows to adjust the relative priority of the checked references. If two libraries describe items with the same name, the item described in the higher-listed library is used by your script.
4. Click OK when finished adding type library references to your script.

The Object Browser

The **ActiveX Automation Members** dialog shows the methods and properties of the objects available to your script. This dialog provides a quick way to check the names of objects or to determine the argument list for a method.

Choose the **Debug | Browse** command to open the **ActiveX Automation Members** dialog.



Click **Debug | Browse** in *Scripter* to open the **ActiveX Automation Members** dialog.

- The *Library* list shows object libraries available for use in the current script. These are the same libraries checked in the **References** dialog. See the *Type Library References* section of this chapter for information about adding type library references in your script.
- The *Data Type* list shows objects available in the type library selected from the *Library* list.
- The *Methods/Properties* list shows methods and properties available from the object selected in the *Data Type* list.
- The text box along the top of the dialog demonstrates the usage of the selected method or property. Click the *Paste* button to insert this sample into the code window.
- Information about the item selected in the *Methods/Properties* list is shown along the right half of the dialog. The top line shows the selected item's type (function, property, or property get) and its name. The second line, labeled *Result*, shows the type of value returned by the method or property. The third line, labeled *Dispatch ID*, shows a value that uniquely identifies a property of method (this value is not used in scripts).

If a method or property returns a reference to another object in the type library, the *Result* label is replaced by a button labeled *Follow Value*. Click the *Follow Value* button to see the

Chapter 22 - Automating Surfer

definition of the object returned by the method or property. After clicking the *Follow Value* button, the *Back* button is enabled. Click the *Back* button to return to the definition you were previously viewing.

- The *Help String* group shows a short description of the item selected in the *Methods/Properties* list. Click the ? button to view the help file associated with the type library. Not all type libraries have help files available.
- If the item selected in the *Methods/Properties* list is a method that accepts parameters, a *Parameters* list shows the names and types of all parameters used by the method.
- Click the *Close* button to dismiss the object browser dialog.

Compatibility with Visual BASIC

The **Scripter** BASIC programming language is compatible with the Visual BASIC for Applications language (VBA). Scripts that run in **Scripter** will work in a VBA environment with few or no modifications. **Scripter** programs will also work under Microsoft Visual BASIC. Unlike most Visual BASIC programs, however, **Scripter** programs are not event-driven. **Scripter** programs are procedural. They start with the first statement of the `Main` procedure, and end when the `Main` procedure ends.

Some statements available in VBA are not supported in **Scripter** BASIC:

The VBA <i>Collection</i> object	All Financial functions	LinkExecute
The VBA <i>Clipboard</i> object	Resume at current line	LinkPoke
GoSub	Erl	LinkRequest
On...GoSub	Option Compare	LinkSend
On...Goto	Conditional compilation	Line numbers
GoSub...Return	With Events	LoadPicture

Multiple statements on one line (separated by " : ")

Conversely, some features of the **Scripter** BASIC language are not supported by VBA. Do not use the following features if you want to transfer your scripts from **Scripter** into VBA:

Clipboard function	CallersLine	DDEExecute
User dialogs	PortInt	DDEPoke
MacroDir	Wait instruction	DDERequest
MacroRun	DDEInitiate	DDETerminateAll
MacroRunThis	DDETerminate	

Suggested Reading

This section has briefly introduced how to write scripts with the **Scripter** programming language. More information is available from **Help | BASIC Language Help**. For additional help in learning how to program or for more information about the Visual BASIC for Applications (VBA) language, which is nearly identical to the **Scripter** BASIC language, we recommend the following books:

Harris, Matthew (1997), *Teach Yourself Visual BASIC 5 for Applications in 21 Days*, Third Edition, SAMS Publishing, Indianapolis, IN, 1248 pp.

Lomax, Paul (1998), *VB and VBA in a Nutshell: The Languages*, O'Reilly & Associates, Inc., Sebastapol, CA, 633 pp.


Wang, Wallace (1998), *Visual BASIC 6.0 for Windows for Dummies*, IDG Books Worldwide, Foster City, CA, 477 pp.


Running Scripts

Scripts are nothing but plain text files. You could create script files with any text-editing program, such as the Windows Notepad. The usefulness of scripts is not realized until the script instructions are carried out. The **Scripter** program is designed to interpret script instructions and to carry them out. The process of carrying out the instructions in a script is called running or executing the script.

Running Scripts from Scripter

Scripts are placed in the code window by typing a new script from scratch or by loading the script with the **File | Open** command. To run the script in the **Scripter** code window, select the **Script |**

Run command, press the F5 key, or click on the  button. **Scripter** examines the script instructions, and, if all the instructions are recognized, it begins to perform each instruction in turn.

Select the **Script | End** command or click the  button to stop executing a script. This may be necessary when you want to edit a script after a run-time error occurs, or when you accidentally start a script and you want to cancel the execution.

Running Scripts from the Command Line

You can run scripts from a command prompt without having to manually load and execute the script in **Scripter**. The same commands that you would type at a command prompt may also be entered as the target for a shortcut in order to link a shortcut icon to a script. Enter the following to run a script from the command line or to link a shortcut to particular script file:

```
<Scripter path> -x filename.bas
```

where `<Scripter path>` represents the path to the **Scripter** program file (for example, `C:\ProgramFiles\Surfer\Scripter\Scripter.exe`), and `filename.bas` represents the name of the script to run. The space between the `-x` and the file name is required. This command opens the **Scripter** window, loads the specified script file, and runs the script. When the script terminates - either successfully or unsuccessfully - the **Scripter** window closes.

To load a script file but not execute it, the following command can be used:

```
<Scripter path> filename.bas
```

This opens the **Scripter** window and loads the specified script file. The **Scripter** window remains open, allowing you to work with the script.

Debugging Scripts

Bugs are errors in a script that keep it from performing as intended. Debugging is the process of locating and fixing these errors. The most common bugs are typographical errors in scripts or malformed instructions. **Scripter** immediately detects these types of errors when you try to run a script. The program beeps, highlights the line containing the error in red (or whatever color has been set with **View | Colors** command), and displays an error message on the status bar.

Viewing Errors

Before running a script, verify that the **View | Status Bar** command is enabled, otherwise you will not see the error message. To resolve the errors that **Scripter** detects immediately, you usually must interpret the error message and correct the indicated problem. Typical errors are typing mistakes, unbalanced parentheses, misuse of a BASIC language instruction, or failure to declare variables in a `Dim` statement (if you use the `Option Explicit` statement). If you do not see an obvious problem, refer to the online BASIC language help to make sure you are using the right syntax.

Run-Time Errors

Scripts that encounter errors midway through script execution may be fixed much the same way as syntax errors. The error message should guide your actions. Some run-time errors cannot be detected until they are executed, such as when you try to open a file that does not exist. In these cases, you need to check for error conditions in your scripts. For example, use the `Dir` function to make sure a file exists before trying to open it. Alternatively, you can use the `On Error GoTo` statement to specify an error handling section to execute when a procedure encounters a run-time error:

```
Sub OpenFile(srf As Object, filename As String)
    On Error Goto ErrLabel
    srf.Documents.Open filename
Exit Sub

ErrLabel:
    MsgBox "Unable to open file " & filename
    Exit ' Must use RESUME or EXIT at end of error handling code
End Sub
```

Script Runs Incorrectly

Most difficult to correct are scripts which run, but do not work as expected. Fixing these scripts is hard because you do not know which line or statement is causing the problem. **Scripter** provides a number of debugging features to help you locate the source of problems.

Debug.Print


Probably the simplest debugging technique is to insert instructions into your script to monitor the progress of the script and display the values of variables at various points throughout the script. Use the `Debug.Print` statement to display information in the **Scripter** immediate window:

```
Debug.Print "The value of variable x is "; x
```

To clear the contents of the immediate window, select the text in the window and press either `DEL` or `BACKSPACE`.


Stop or Pause

Insert the `Stop` instruction to pause script execution where you think there might be a problem. While the script is paused, you can examine and change the values of program variables. If a running script appears unresponsive, it may be stuck in an infinite loop. Select the **Script | Pause**

command or click the  button to pause the script. To resume executing a paused script, select the **Script | Run** command.

Viewing Variable Values

While a script is paused, there are several ways to view the value of a variable:

- In the immediate window, type a question mark, followed by the variable name, and the press ENTER. The current value of the variable is displayed.
- In the code window, place the cursor on the variable name you want to examine (that is, click on the variable name in the code window). Press SHIFT+F9, select the **Debug | Quick Watch** command, or click the  button. The current value of the variable is displayed in the immediate window.
- To continuously monitor a variable's value, click on the variable name in the code window, and press CTRL+F9 or select the **Debug | Add Watch** command. Alternatively, type the variable name in the watch window and press ENTER. The variable name and its value are displayed in the watch window. Every time script execution pauses, the variable value is automatically updated. To clear a variable from the watch window, highlight the line showing the variable value and press the DEL or BACKSPACE key.

Changing Variable Values

To change the value of a variable, type an assignment expression in the immediate window and press ENTER. For example, type "A=5" (without quotes) and press ENTER to assign a new value to the variable named "A."

Step

A powerful debugging technique is to watch **Scripter** execute your script one line at a time. This lets you check the effect of each instruction and verify that the script is doing what you expect. While stepping through a script, you can examine and change the values of script variables. Select the **Script | Run** command to resume script execution at full speed after stepping through script instructions.

- To execute your script one line at a time press the F8 key, or select the **Debug | Step Into** command. The first line of the script is executed (or, if the script was paused, the next

highlighted line is executed). The next line is highlighted and a yellow arrow appears to the left of the next line. To execute the highlighted instruction, press F8 again.

- If a statement calls a subroutine or function that is defined within your script, the highlight will move into the called procedure. To keep from tracing execution into a called procedure, press SHIFT+F8 or select the **Debug | Step Over** command. This executes the whole subroutine or function in a single step.
- If you accidentally step into a procedure, press CTRL+F8 or select the **Debug | Step Out** command. This executes all remaining instructions in a procedure, and returns the highlight to the instruction that called the procedure.
- If you do not see the next highlighted instruction, select the **Debug | Show Next Statement** command to scroll the highlighted line into view.
- Sometimes you may want to skip the execution of some instructions or you may want to execute the same instructions several times without restarting the script. To change the next instruction line, click on the line you want to execute next and select the **Debug | Set Next Statement** command.

Breakpoint

Watching **Scripter** execute every line of the script may be too time consuming. A breakpoint is a line of code that you mark and this pauses the script where you think there might be a problem. When **Scripter** encounters a line marked as a breakpoint, it pauses the script just as if it had executed a `Stop` instruction. Breakpoints are more convenient than `Stop` instructions because they may be set and cleared while a script is paused, whereas `Stop` instructions may be changed only after a script has ended.

- To set a breakpoint, click in the break bar area next to the line you want to mark. The break bar is the area to the left of the code window, between the sheet tabs and the code window (see page 510). Alternatively, click on the line you want to mark, and press F9 or select the **Debug | Toggle Break** command. The line becomes highlighted in red, and a round marker appears in the break bar area.
- To clear a breakpoint, click on the round marker, or move the cursor to the marked line and press F9 or select the **Debug | Toggle Break** command again. Clear all breakpoints by pressing SHIFT+CTRL+F9 or selecting the **Debug | Clear All Breaks** command.

A quick alternative to setting a breakpoint is the **Debug | Step to Cursor** command. This command has the same effect as setting a breakpoint on the current line, running the script, and then clearing the breakpoint after script execution has paused on the current line.

Trace

To check flow of execution through your script without having to watch each line of the script being executed, try using the `Trace` function. To activate the trace function type "Trace" (without the quotes) in the immediate window and press ENTER. `Trace On` is displayed in the immediate window. As the script is run, the location of every instruction being executed is printed in the immediate window. After the script finishes, the trace function is automatically disabled.

Stack

If you nest procedure calls (that is, one procedure calls another procedure, which calls yet another procedure, and so forth), the stack window may be useful. When a script is paused, the stack window lists the procedures that have been called, and the order in which they were called. For instance, if the `Main` procedure calls procedure A which in turn calls procedure B, the stack window displays three lines, one for each of the called procedures. Clicking on a line in the stack window moves the corresponding procedure into view in the code window.

Modules

Click the loaded window tab in the immediate window area to see which module files are currently being interpreted by **Scripter**. The loaded files include the current script file and any modules it includes with the `' #Uses` statement.

Examples

This section provides several examples that illustrate some of the more common tasks in **Surfer** automation. Please see the online help and the `SAMPLES` folder for additional examples.

Creating and Printing a Contour Map

This example automates the process of creating a contour map to visualize changes in groundwater levels. The script prompts the user for a data file, creates a grid from the file, creates a contour map, prints the map, and saves the map in a **Surfer** file [`.SRF`].

```
Sub Main
    ' Declare object and string variables used in the script
    Dim SurferApp, Plot, ContourMapFrame, ContourMap As Object
    Dim InFile, GridFile, BaseName As String
```



```
' Create the Surfer Application object and assign it to the "SurferApp"
' variable
Set SurferApp = CreateObject("Surfer.Application")
SurferApp.Visible = True      ' Make Surfer visible

' Prompt the user for the name of the data file to process.
InFile = GetFilePath("", "DAT;TXT;CSV;XLS", CurDir(), "Select data file", 0)
If InFile = "" Then End      ' Cannot continue: no file was selected

' Get file name without the extension (for example, "week5.txt" becomes "week5")
BaseName = InFile
ExtStart = InStrRev(InFile, ".")
If ExtStart > 1 Then BaseName = Left(InFile, ExtStart - 1)

' Create a grid from the specified data file using the kriging algorithm
GridFile = BaseName + ".grd"
SurferApp.GridData DataFile:=InFile, Algorithm:=srfKriging, _
    DupMethod:=srfDupNone, ShowReport:=False, OutGrid:=GridFile

' Create a plot document in Surfer and assign it to the variable named "Plot"
Set Plot = SurferApp.Documents.Add(srfDocPlot)

' Create a contour map. Assign the map frame to the "ContourMapFrame" variable
Set ContourMapFrame = Plot.Shapes.AddContourMap(GridFile)

' Assign the contour map attributes to the variable named "ContourMap"
Set ContourMap = ContourMapFrame.Overlays(1)

ContourMap.FillContours = True      ' Fill the contour map levels
Plot.PrintOut                      ' Print the page on the default printer
Plot.SaveAs(BaseName + ".srf")      ' Save the map
End Sub
```

Opening, Saving, and Closing Documents

The *Documents* collection provides access to the **Surfer** file commands. Use the *Documents* object's *Open* method to open an existing plot, worksheet, or grid file. Use the *Add* method to create a blank plot or worksheet. The *SaveAll* method saves all open documents, and the *CloseAll* method closes all open documents. To save or close an individual document, use the *Document* object's *Save*, *SaveAs*, and *Close* methods.

```
Sub Main
    Set srf = CreateObject("Surfer.Application")
    srf.Visible = True

    ' Create a blank plot
    srf.Documents.Add srf.DocPlot

    ' Create a blank worksheet
    srf.Documents.Add srf.DocWks

    ' Open an existing plot
    filename$ = GetFilePath("SRF")
    If filename$ <> "" Then
        srf.Documents.Open filename$
    End If

    ' Open the sheet named "Sheet1" from an Excel file
    filename$ = GetFilePath("XLS")
    If filename$ <> "" Then
        srf.Documents.Open filename$, "Sheet=Sheet1"
    End If

    ' Close the active document
    srf.ActiveDocument.Close

    ' Save the active document using its current name
    If Not srf.ActiveDocument.Saved Then
        srf.ActiveDocument.Save
    End If
```

```
' Save the document whose window caption is "Plot1"
srf.Documents("Plot1").SaveAs "MyDocument"

' Close all documents
srf.Documents.CloseAll
End Sub
```

Creating a Variogram

The *Variogram* object represents a variogram. A *Variogram* object is a type of *Shape* object. Create a new *Variogram* using the *Shapes* collection's *AddVariogram* method. The *AddVariogram* method returns a reference to a *Variogram* object. Use the properties and methods of the *Variogram* object to access information about the computed variogram and to modify the options.

A *VarioComponent* object represents one component of a variogram model. When you use the kriging algorithm to create a grid file, you can specify the variogram model to use by passing an array of *VarioComponent* objects to the *Application* object's *GridData* method. You can create a *VarioComponent* by calling the *Application* object's *NewVarioComponent* method. Use the *Variogram* object's *Model* property to obtain the array of *VarioComponent* objects used by a variogram.

The following subroutine creates a *Variogram* object, and then creates a grid file using the computed variogram components.

```
Sub GridWithComputedVariogram(plot As Object, datafile As String, grid As String)
' Create a variogram.
Set v = plot.Shapes.AddVariogram (datafile, ShowReport:= False)

' Create a grid using the computed variogram
plot.Application.GridData DataFile:=datafile, Algorithm := srfKriging, _
    KrigVariogram := v.Model, _
    ShowReport := False, OutGrid := grid

' Delete the variogram
v.Delete
End Sub
```

Chapter 22 - Automating Surfer

The next subroutine creates two *VarioComponent* objects and creates a grid based on these variogram components.

```
Sub GridWithStaticVariogram(plot As Object, datafile As String, grid As String)
    ' Get the Application object
    Set srf = plot.Application

    ' Create two VarioComponent objects and store them in an array
    Dim components(1 To 2) As Object
    Set components(1) = srf.NewVarioComponent(srf.VarNugget,1.0,2.0,3.0)
    Set components(2) = srf.NewVarioComponent(srf.VarLinear,1.0,2.7,1.5)

    ' Create a grid using the two variogram components
    srf.GridData DataFile := datafile, Algorithm := srf.Kriging, _
        KrigVariogram := components, _
        ShowReport := False, OutGrid := grid
End Sub
```

Finally, the following Main procedure demonstrates how you might use the preceding subroutines:

```
Sub Main
    ' Start Surfer, set the default directory, and create a plot document
    Set srf = CreateObject("Surfer.Application")
    srf.Visible = True
    srf.DefaultFilePath = srf.Path + "\samples"
    Set plot = srf.Documents.Add(srf.DocPlot)

    ' Create a grid using a pre-determined variogram model
    GridWithStaticVariogram plot, "demogrid.dat", "VarioDemo1"

    ' Create a grid using a computed variogram model
    GridWithComputedVariogram plot, "demogrid.dat", "VarioDemo2"

    ' Create contour maps to compare the two grids
    plot.Shapes.AddContourMap "VarioDemo1"
    plot.Shapes.AddContourMap "VarioDemo2"
End Sub
```

Overlaying Maps

To overlay maps in a script, you must first select the map or *MapFrame* objects to be overlaid, and then call the *Selection* collection's *OverlayMaps* method. The following subroutine demonstrates how to overlay maps:

```
Sub OverlayMapsExample(mapframe1 As Object, mapframe2 As Object)
    ' Retrieve the parent PlotDocument object
    Set plot = mapframe1.Parent

    ' Clear all selections and then select the two MapFrame objects
    plot.Selection.DeselectAll
    mapframe1.Selected = True
    mapframe2.Selected = True

    ' Overlay the selected maps
    plot.Selection.OverlayMaps
End Sub
```

The subroutine would be used like this:

```
Sub Main
    ' Start Surfer and create a blank plot document
    Set srf = CreateObject("Surfer.Application")
    srf.Visible = True
    Set plot = srf.Documents.Add(srf.DocPlot)

    ' Create a contour map and a vector map
    gridfile$ = srf.Path + "\samples\demogrid.grd"
    Set map1 = plot.Shapes.AddContourMap(gridfile$)
    Set map2 = Plot.Shapes.AddVectorMap(gridfile$)

    ' Call the example subroutine to overlay the maps
    OverlayMapsExample map1, map2
End Sub
```

Modifying Axes

Axis settings may be changed using the *Axis* object. You access *Axis* objects through the *Axes* collection object, which in turn is accessed through a *MapFrame* object. The following example does not use subroutines, but you could write the script with subroutines instead.

```
Sub Main
    ' Start Surfer and create a blank plot document
    Set srf = CreateObject("Surfer.Application")
    srf.Visible = True
    Set plot = srf.Documents.Add(srf.DocPlot)

    ' Create a contour map
    gridfile$ = srf.Path + "\samples\demogrid.grd"
    Set mapframe = plot.Shapes.AddContourMap(gridfile$)

    ' Turn off screen redrawing to speed up this procedure
    old_update_setting = mapframe.Application.ScreenUpdating
    mapframe.Application.ScreenUpdating = False

    ' Adjust the left and bottom axes
    For Each axis In mapframe.Axes
        If axis.AxisType = srf.ATLeft Or axis.AxisType = srf.ATBottom Then
            With axis
                .MajorTickType      = srf.TickCross
                .MinorTickType      = srf.TickIn
                .MinorTickLength    = 0.1
                .MinorTicksPerMajor = 9
                .ShowMajorGridLines = True
                .AxisLine.Width     = 0.03
            End With
        End If
    Next

    ' Restore the previous screen setting
    mapframe.Application.ScreenUpdating = old_update_setting
End Sub
```

Chapter 23

Preferences

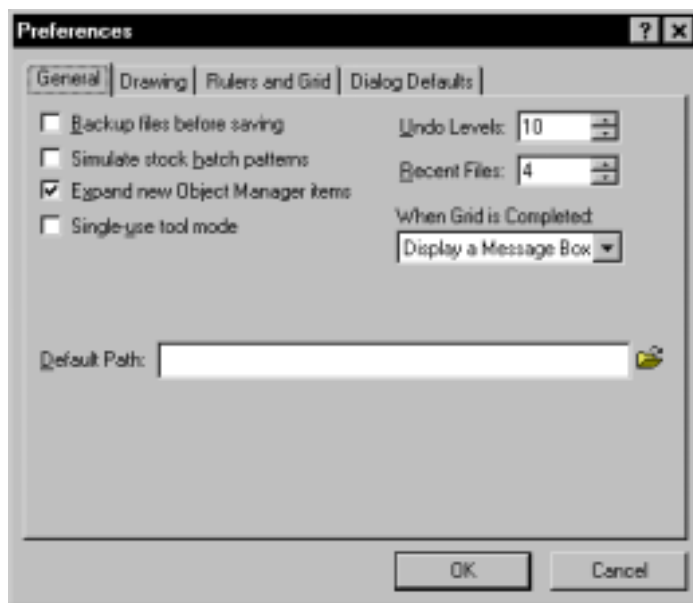
Introduction to Preferences

Surfer's default settings are customized by selecting the **File | Preferences** menu command when a plot window is active. By adjusting the settings in the **Preferences** dialog, you can customize **Surfer** to suit your individual preferences and work habits. The settings are automatically saved and restored whenever **Surfer** is restarted.

General

If *Backup files before saving* is checked, a backup copy of an existing file is created before saving the document. The backup copy is saved with a [.BAK] extension. If an identically named backup file already exists, it will be overwritten.

Check the *Simulate stock hatch patterns* option to force **Surfer** to draw Windows stock fill patterns as a sequence of evenly spaced parallel lines. Remove the check mark to use the normal Windows stock patterns. Windows stock patterns include *Diagonal Cross*, *Crosshatch*, *Forward Slash*, *Backward Slash*, *Vertical*, and *Horizontal* and they are noted as *Type: Stock* in the fill pattern palette. Simulated hatch patterns maintain the same distance between lines no matter what the resolution of the output device. Non-simulated patterns are resolution dependent and become closer



The **General** page of the **Preferences** dialog allows you to customize many of the plot window features.

Chapter 23 - Preferences

together when output on most printers. The disadvantage of simulated lines is that they are not as aesthetic on the screen and take significantly longer to draw. If the spacing of the hatch lines in printed output is not important, leave this option turned off.

Check *Expand new Object Manager items* to force the **Object Manager** to expand all new object hierarchies in the tree control. If this option is turned off, new parent objects will contain a + icon. The + icon can be clicked to manually expand additional levels.


Check *Single-use tool mode* to make the toolbar items active one time. After using a tool once (for example, drawing a polygon with the polygon tool), the current tool reverts to the selection tool. If this box is not checked, the tools are persistent until another tool is selected or ESC is pushed on your keyboard.

The number of actions remembered for the **Edit | Undo** command is set in the *Undo Levels* box. Once the maximum number of actions has been performed, the oldest action is dropped off the list as new actions are added. The maximum number of undo levels is 100. Undo can consume significant amounts of memory, so this option should probably be left somewhere between 3 and 10 if memory is at a premium. Set the undo levels to 0 to disable **Undo** completely.

The *Recent Files* option controls the number of recently used files listed near the bottom of the **File** menu. Set this to a value between 0 and 16.

There are three notification options you can choose from in the *When Grid is Completed* list: *Do Nothing*, *Play a Sound*, or *Display a Message Box*. This action is performed after most gridding operations to inform you that the operation has completed.

Enter a path into the *Default Path* box to set the default path for opening and saving files.

Alternatively, click the  button to browse for a path. This option sets the initial directory displayed in the **File | Open** and **Save** dialogs after starting **Surfer**.

Drawing

The **Drawing** page in **File | Preferences** controls selection, drawing properties (e.g. default line color), redraw, page and margin display, and page units.

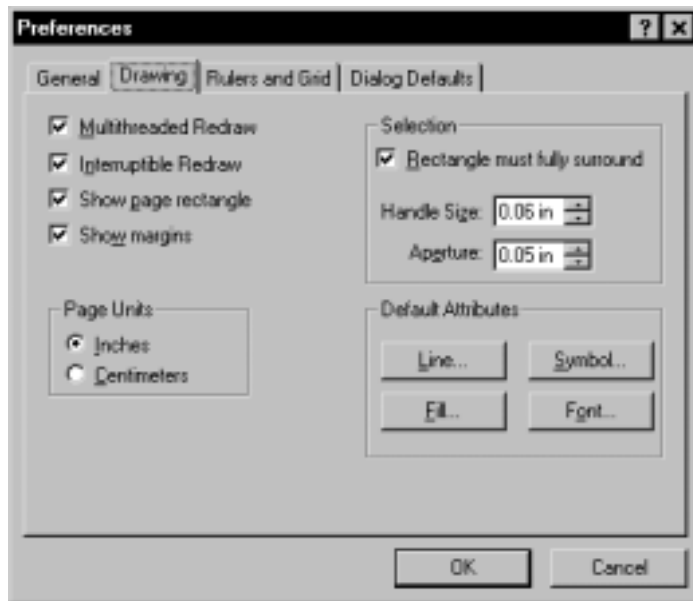
Selection Group

Use the options in the *Selection* group to control how objects are selected in the plot window.

The *Rectangle must fully surround* option controls how select functions. If the box is checked, the selection rectangle must be drawn completely around the object to select it. If any portion of the object extends beyond the selection rectangle, the object is not selected. If this option is unchecked, the selection rectangle only needs to partially intersect an object to select it.

The *Handle Size* box controls the width and height of the green selection handles that appear around selected objects.

The *Aperture* box controls the distance between the pointer and the object when selecting an object. The smaller the aperture, the closer the pointer must be to the object's bounding box to select the object. When the aperture is set to zero, the pointer must be directly on the object to select it. The aperture setting range is from 0.0 inches to 0.50 inches. The aperture size does not affect the size of the bounding box, it just controls the distance the pointer can be from the object to select it.



*The **Drawing** page of the **Preferences** dialog allows you to control drawing features.*

Default Attributes Group

The *Default Attributes* group specifies the default drawing object properties used when creating new objects.

- *Line* properties are used to specify the default line style, color, and width. Some examples of where this is used include axes, stand-alone lines, and the outer border of rectangles, ellipses, and polygons.
- *Fill* properties specify the fill pattern and color used for the interior of polygons, rectangles, and ellipses.
- *Symbol* properties include the symbol set, color, and size. These properties are used for stand-alone symbols, post maps, and imported graphics that do not use a specific symbol.
- *Font* properties specify the default face, style, size, color, and text alignment. Some examples of where this is used include stand-alone text, axis labels, post map labels, and inline contour labels.

For more information on setting the individual properties in the **Line**, **Fill**, **Symbol**, or **Text Properties** dialogs, see the online help.

Miscellaneous Options

Multithreaded Redraw allows redraws in the background while doing things like scrolling, zooming, and clicking on menu commands, etc. On a multiprocessor system, the user interface thread runs on one processor and the screen redraw runs on the second processor at the same time. On a single processor system, the two threads are executed by quickly switching back and forth between the two. If you are experiencing problems while the plot window is being updated, you might try turning this feature off.

Interruptible Redraw allows other commands to be executed while the plot window is being updated. This is useful when a complicated plot takes a long time to redraw and you wish to make changes before it has completed. After the new commands have been executed, the redraw will start over. Some video cards and printer drivers have problems with this feature. If you are experiencing problems while the plot window is being updated, you might try turning this feature off.

Use the *Show page rectangle* option to turn on or off the outline of the page in the plot window. This is useful to see where the drawing objects will appear on the printed page. Check the *Show margins* box to show margin settings on the page as dashed lines. The margins are set through the **File | Page Setup** menu command.

Page Units are the units used to measure distances on the printed page. This can be set to *Inches* or *Centimeters*.

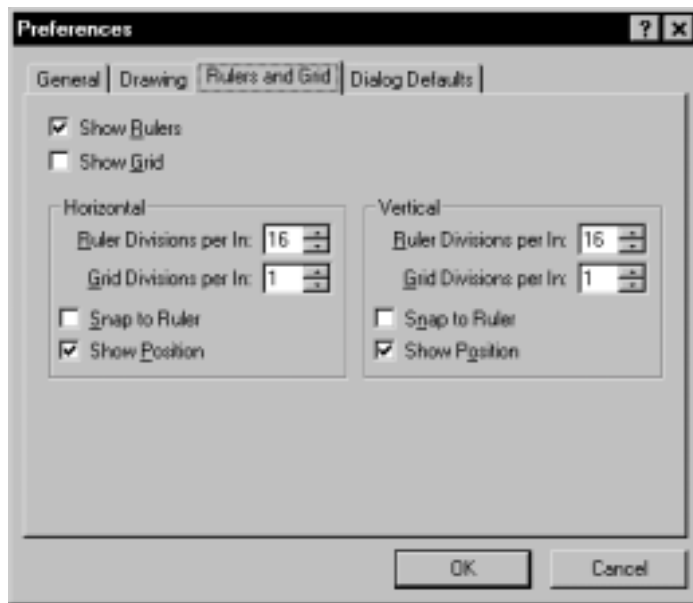
Rulers and Grid

Rulers appear along the top and left edge of the plot window, and are used to position and align objects. Rulers use the current page units. To change the units, see the *Drawing* section of this chapter.

The drawing grid is a series of evenly spaced dots, similar to graph paper, which is used to help align objects in the plot window. A shortcut to the **Rulers and Grid** page is provided by double-clicking either ruler.

Show Rulers turns the horizontal and vertical rulers on and off in the plot window. *Show Grid* turns the drawing grid on and off in the plot window. The horizontal and vertical rulers and grid settings can be set independently.

Use *Ruler Divisions per In* (or *cm*) to set the number of divisions per page unit on the ruler. For example, if *Page Units* is set to *Inches* and this option is set to 16, then rulers will contain 16 divisions per inch. Use *Grid Divisions per In* (or *cm*) to set the number of divisions per page unit on the grid.




The **Rulers and Grid** page aids object placement and alignment.

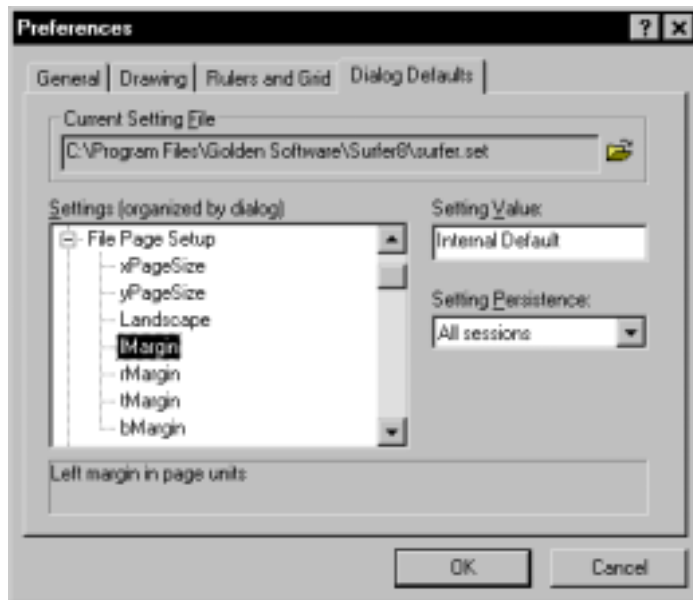
Snap to Ruler causes objects to snap to the ruler divisions as objects are drawn or moved. This allows objects to be easily aligned with the ruler division marks.

Check the *Show Position* option to indicate the current pointer position as gray lines on the rulers. As the mouse is moved, the position indicator moves within the rulers to show the exact page position of the pointer.

Dialog Defaults

The **Dialog Defaults** page controls most of the default settings used throughout **Surfer's** dialogs. These settings are saved within a setting file [.SET], and are reloaded whenever **Surfer** is started. Using the hierarchical list in the **Dialog Defaults** page, you can customize each default setting independently. This is an advanced feature, and the settings should not be changed indiscriminately. Minimal error checking is performed, and it is possible to specify values completely inappropriate for some data sets.

The *Current Setting File* is the file **Surfer** is currently using to extract the default settings. Any changes made to the default settings in the various dialogs are saved in the current setting file when **Surfer** is closed. By default, **Surfer** uses a file named SURFER.SET. It is also possible to specify a different setting file by clicking the  button. See the *Using Custom Setting Files* section later in this chapter for more information on creating alternate setting files.



*The **Dialog Defaults** page allows customization of the default values that appear in other dialogs.*

The *Settings* list displays the default settings organized hierarchically by dialog. To change a setting, click the + icon next to the dialog you wish to modify. Individual settings can be highlighted by clicking on them or by using the up and down arrow keys on your keyboard. As each individual setting is highlighted, a brief description appears in the text box at the bottom of the dialog.

You can enter the new value of the default setting in the *Setting Value* control. This control changes depending on the type of setting highlighted in the *Settings* list. Settings with a discrete number of choices are displayed in a drop-down list. The values entered in this control are not checked for errors, so be sure to specify an appropriate value for the highlighted setting.

All settings have a value called *Internal Default*. This is a special value that indicates that **Surfer** should use whatever it thinks is best for the current setting. If the *Setting* control is a box, you can specify the *Internal Default* value by deleting the contents of the box so it is empty.

Several default settings require the value to be specified in a particular format. The type of format required is determined from the comments at the bottom of the **Dialog Defaults** page. See the *Formats for Attribute Values* section in this chapter for a list of these special values.

The *Setting Persistence* option specifies how the setting is updated when the user changes the setting during normal operation. The options are:

<i>Always Reset</i>	Do not update the default setting when it is changed in a dialog. Every time the dialog is invoked, the setting is reset to the value in the setting file.
<i>Current Session Only</i>	This option saves changes made to the setting within the dialog during the current session only. The settings are not written to the setting file and are not used the next time Surfer is started.
<i>All Sessions</i>	This option saves the changes made to the setting within the dialog during the current session, and writes the changes to the setting file to be used the next time Surfer is started.

You must click the OK button in the dialog that uses the setting for the setting to be updated. If you *cancel* the dialog, the changes are not recorded in the setting file.

Formats for Attribute Values

Several settings use specially formatted values to specify all the required information. When specifying the values for these settings, you must include all the parameters as discussed below.

Colors

Colors are specified by name as they appear in the dialog color palettes. *Appendix D, Line Styles, Fill Patterns, and Colors* contains a list of all color names. The names must be enclosed in double quotes, for example, "Ocean Green."

Alternatively, a color is specified using the special syntax:

"Rxxx Gyyy Bzzz"

where xxx, yyy, and zzz specify a red, green, and blue color component respectively. Each color component can range from 0 to 255.

For example:

"R0 G0 B0" All components are 0, resulting in black

"R0 G255 B0" Pure green

"R255 G255 B255" All components are at full intensity, resulting in bright white

Line Properties

Line properties are specified using the syntax:

"Color" "Style" Width

Parameter	Description
Color	The line color name. See the <i>Colors</i> section in this chapter for details.
Style	The style of the line as it appears in the dialog line style palette or in <i>Appendix D, Line Styles, Fill Patterns, and Colors</i> . The name must be enclosed in double quotes.
Width	The width of the line in thousandths of an inch (mils). This should not be enclosed in quotes.

Example:

"R0 G255 B50" ".1 in. Dash" 10

Fill Properties

Fill properties are specified using the syntax:

"ForeColor" "BackColor" "Pattern" Mode

Parameter	Description
ForeColor	The foreground color name. See the <i>Colors</i> section in this chapter for details.
BackColor	The background color name. See the <i>Colors</i> section in this chapter for details.
Pattern	The fill pattern name as it appears in the dialog fill pattern palette or in <i>Appendix D, Line Styles, Fill Patterns, and Colors</i> . The name must be enclosed in double quotes.
Mode	The background mode specifies whether vector fills should be opaque (1) or transparent (2). The background mode is specified as a value, and is not enclosed in quotes. Transparent backgrounds are only available for vector and stock fills.

Example:

"Red" "Desert Blue" "Backward slash" 2

Font Properties

Font properties are specified using the syntax:

"FaceName" "Color" Style Size

Parameter	Description
FaceName	The font face name as it appears in the Text dialog. The name must be enclosed in double quotes.
Color	The font color name. See the <i>Colors</i> section in this chapter for details.
Style	<p>Add the following values to get the style you want:</p> <p>None = 0 Bold = 16 Italics = 32 Strikethrough = 64 Underline = 128</p> <p>The style is entered as a number, and is not enclosed in double quotes.</p>
Size	The size of the text in points. This is entered as a number and is not enclosed in double quotes.

Example:

"Braggadocio" "Black" 0 6

Symbol Properties

Symbol properties are specified using the syntax:

"SymSet" "Color" Index Size

Parameter	Description
SymSet	The symbol set or face name as it appears in the Symbol Properties dialog. The name must be enclosed in double quotes.
Color	The symbol color name. See the <i>Colors</i> section in this chapter for details.
Index	The 0-based index of the symbol to use as it appears in the Symbol Properties dialog. This is entered as a number, and does not use double quotes.
Size	The size of the symbol in thousandths of an inch (mils). This is entered as a number and is not enclosed in double quotes.

Example:

"Default Symbols" "Purple" 12 440

Numeric Label Properties

Numeric label properties are specified using the syntax:

Type Digits Style "Prefix" "Suffix"


Parameter	Description
Type	The numeric format. This is entered as a number, and is not enclosed in double quotes. 0 = Fixed 1 = Exponential 2 = Compact
Digits	The number of decimal digits after the decimal place in the label. This value can range from 0 to 15. This is entered as a number, and is not enclosed in double quotes.
Style	This is entered as a number, and is not enclosed in double quotes. 0 = None (no styles) 1 = Thousands (separate thousands with a comma) 2 = Absolute Value (show numbers as absolute values) 3 = Thousands and Absolute Value (use thousands and absolute values)
Prefix	The Prefix text is text that appears at the beginning of every label. This must be enclosed in double quotes.
Suffix	The Suffix text is text that appears at the end of every label. This must be enclosed in double quotes.


Example:

0 8 0 "Pre" "Post"

Using Custom Setting Files

To create a custom setting file:

1. make a backup of the default SURFER.SET file,
2. click on **File | Preferences** and then click on the **Dialog Defaults** page,
3. open the setting file by clicking the  button next to the *Current Setting File* field,
4. modify the setting file on the **Dialog Defaults** page, and then
5. save the changes to the file by clicking OK in the **Preferences** dialog.

After the preferences are set, you can rename the set file, keeping the [.SET] extension. Each setting file can use a different set of defaults. You can manually change the setting file **Surfer** uses in the **Dialog Defaults** page of the **Preferences** dialog by clicking the  button next to the *Current Setting File* field.

Alternatively, to use a different setting file with **Surfer**, start **Surfer** with the /SET switch on the command line, as in:

```
Surfer.exe /SET SetFile
```

where `SetFile` is the path and name of an existing setting file.

To make this entire process easier, you can create a new shortcut and specify the /SET switch in the *Target:* field of the shortcut properties dialog. See your Windows documentation for details on how to create shortcuts.

You can also use old **Surfer 6** setting files in **Surfer 8**. When using **Surfer 6** setting files, the hierarchy displayed in the **File | Preferences** dialog is only a single level due to lack of nesting information in the old format. **Surfer 7** setting files can be used in **Surfer 8** without any changes to the [.SET] file.

Appendix A

Mathematical Functions

Introduction to Mathematical Functions

Mathematical Functions are used to modify data with the **Grid | Function**, **Grid | Math**, **Grid | Data**, and **Grid | Variogram** commands (plot window), and with the **Data | Transform** command (worksheet). For more information on creating a grid with **Grid | Function** or using the data filters with **Grid | Data**, see *Chapter 4, Creating Grid Files*. Using the data filters with **Grid | Variogram** is discussed in *Chapter 5, Variograms*. **Grid | Math** is discussed in *Chapter 18, Grid Operations*. Use the **Data | Transform** command to apply mathematical transformations to worksheet columns. See the *Transforming Data in the Worksheet* section in this chapter and see *Chapter 3, Data Files and the Worksheet* for more information on **Data | Transform**.

The mathematical expression can consist of constants, variables (such as column letters), or functions (outlined below). The formulas follow standard precedence rules. Spaces are used in the equation for clarity. Equations can be up to 256 characters, including spaces, in length.

Formula Calculation

Formulas are specified using standard precedence rules. Operators, in order of decreasing precedence, are:

- () parentheses
- unary minus (or negative sign)
- * / multiplication and division
- + - addition and subtraction

Operators of equal precedence are evaluated from left to right within the equation. Parentheses are used to override precedence, and expressions within the parentheses are performed first.

Appendix A - Mathematical Functions

Mathematical Functions

The following built-in functions are supported.

Trigonometric Functions

All trigonometric functions are carried out in radians. If the data are in degrees, use the `d2r(x)` conversion function (under *Miscellaneous Functions*) to convert degree data to radians and then use the trigonometric functions.

<code>sin(x)</code>	Sine
<code>cos(x)</code>	Cosine
<code>tan(x)</code>	Tangent, x must not be an odd multiple of $\pi/2$
<code>asin(x)</code>	Arcsine in the range $-\pi/2$ to $\pi/2$, x must be between -1 and 1
<code>acos(x)</code>	Arccosine in the range 0 to π , x must be between -1 and 1
<code>atan(x)</code>	Arctangent in the range $-\pi/2$ to $\pi/2$
<code>atan2(y,x)</code>	Arctangent in the range $-\pi$ to π

Bessel Functions

<code>j0(x)</code> <code>j1(x)</code> <code>jn(n,x)</code>	Bessel functions of the first kind of orders 0, 1, and n
<code>y0(x)</code> <code>y1(x)</code> <code>yn(n,x)</code>	Bessel functions of the second kind of orders 0, 1, and n. x must be greater than or equal to zero

Exponential Functions

exp(x)	Exponential function (e to the x)
sinh(x)	Hyperbolic sine
cosh(x)	Hyperbolic cosine
tanh(x)	Hyperbolic tangent
ln(x)	Natural logarithm, x must be positive
log10(x)	Base 10 logarithm, x must be positive
pow(x,y)	X raised to the y th power. Error conditions result if: x is zero and y is negative or zero, x is negative and y is not an integer.

Miscellaneous Functions

min(x,y)	Smaller of x and y
max(x,y)	Larger of x and y
randn(x,y)	An approximately normally (Gaussian) distributed pseudo-random number with mean x and standard deviation y
randu(x)	A uniformly distributed pseudo-random number from the interval [0,x]
row()	Returns the row number
ceil(x)	Smallest integer greater than or equal to x
floor(x)	Largest integer less than or equal to x
sqrt(x)	Square root of x, x must be greater than or equal to zero
fabs(x)	Absolute value of x
fmod(x,y)	Floating point remainder of x/y, if y is zero, fmod returns zero
d2r(x)	Convert values in degrees to radians
r2d(x)	Convert values in radians to degrees. Example: sin(d2r(30)) computes the sine of 30 degrees. Sin(30) computes the sine of 30 radians.

Appendix A - Mathematical Functions

Statistical Functions of an Interval of Columns

sum(a..z)	Calculates the sum of a range of columns in a row
avg(a..z)	Calculates the average of a range of columns in a row
std(a..z)	Calculates the (population) standard deviation of a range of columns in a row
rowmin(a..z)	Finds the minimum value of a range of columns in a row
rowmax(a..z)	Finds the maximum value of a range of columns in a row

The *Statistical Functions of an Interval of Columns* functions operate row-wise on an interval of columns. For example, SUM(A..Z) computes the sum of the twenty-six columns A, B, C, ..., Z. It does this for each row separately. Replace 'a..z' by any valid interval of columns, such as 'C..H' or 'W..AC'. There must be exactly two periods between the column labels. Columns may be given in reverse order, such as SUM(Z..A).

Boolean Expressions

Boolean expressions, include:

- logical operators (and, or, xor, not)
- comparison operators (=, <>, <, >, <=, >=)
- the IF function - IF(condition,true_value,false_value)

The words AND, OR, XOR, NOT, and IF are reserved keywords and may not be used as variable names.

Transforming Data in the Worksheet

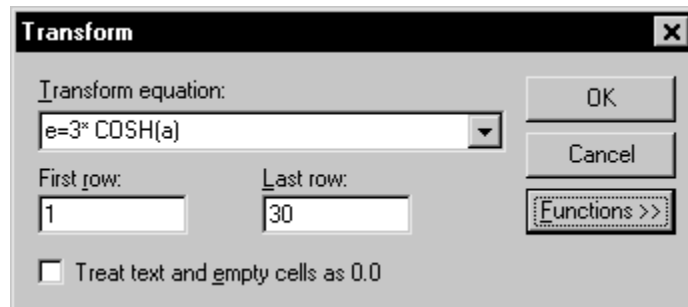
Worksheet equations consist of a destination column letter on the left side of the equation and a mathematical expression on the right side. Only specified rows are included in the calculation. An example of an equation is $C = A + B$. If the *Transformation Rows* option in the **Transform** dialog is set from one to 30, the formula places the sum of the cells in Columns A and B into Column C. The equation is calculated for rows one through 30 only. Only numeric cells are used in calculations.

You can click the *Functions* button in the **Transform** dialog to add pre-defined mathematical functions to the equations. Change the variable (X) in the listed functions to a column letter in the transformation equation.

Example

Let's say you would like to transform the numbers in Column A in the worksheet with the equation $3\cosh(x)$. That is, for numeric values in Column A, find the hyperbolic cosine, multiply by three, and place the results in Column E.

1. Open the data file in the worksheet with **File | Open**.
2. Click the **Data** menu and select **Transform**. A worksheet column does not need to be selected.
3. Delete any equations that appear in the *Transform equation* box.
4. Enter a destination column followed by an equals sign into the *Transform equation* box, in this case, E.
5. Enter 3* following the equal sign.
6. Click the *Functions* button.
7. Select *COSH(X)* from the function list, and then click the *Insert* button. *COSH(X)* is inserted into the *Transform equation* box.
8. Replace the X with the column letter, in this case, A.



The destination column is Column E and the transformed column is Column A. The calculation is made for rows 1 through 30 only.

The transformed numbers appear in the destination column in the rows selected in the **Transform** dialog. Remember that mathematical operators are necessary in the transform equations. If the above equation were entered without the multiplication sign (*), an error message would result.

Appendix A - Mathematical Functions

Errors

Error codes and special numeric values can appear in a worksheet cell depending on the type and nature of the condition within the cell.

Code	Explanation
#####	Number does not fit in the column. The column must be wider for the number to be shown.
#N/A	The value cannot be computed. For example, there is not enough data to calculate a statistic.
#DIV/0!	An attempt to divide-by-zero was made in performing a calculation.
#ERROR	A value could not be computed. For example, the square root of a negative number cannot be computed.
#OVERFLOW	The value is too large for the worksheet. The largest absolute value is about 1.797E+308.
1.#INF	The value is too large for the worksheet (i.e., "infinite" value).
1.#IND	The numeric value is indefinite. This is usually the result of performing a calculation with an infinite value or attempting to divide by zero.

Examples

The following are examples of mathematical function syntax used in **Surfer**. If you use **Data | Transform** in the worksheet, replace X, Y, and Z with column letters.

Equation	Mathematical Function Syntax
x^2	pow(x,2)
$\ln(x)$	ln(x)
$\log_{10} x$	log10(x)
$1 - e^{-x}$	(1-exp(-X))
$1 - e^{-x^2}$	(1-exp(-X*X))
$1 - \frac{\sin(x)}{x}$	1-(sin(x)/x)
$\frac{x^2}{1 + x^2}$	pow(x,2)/(1+(pow(x,2)))
$2x - x^2$	(2 * X)-pow(x,2)
$(x^2 + y^2)(\sin(8 \times \tan^{-1} xy))$	(pow(x,2)+pow(y,2))*(sin(8*atan(x*y)))

Appendix B

Math Text Instructions

Introduction to Math Text

The math text instruction set offers advanced formatting of text in the plot window. Math text instructions can be used to change the typeface, size, color, weight, and style of text on a character-by-character basis. Greek letters and mathematical symbols can be written in the plot window using math text instructions. The math text instructions also allow for the detailed placement of characters and symbols; thus, superscripts, subscripts, and the superposition of characters are possible.

All text within the plot window can be controlled by math text instructions. For example, text blocks can include math text instructions via the **Text** dialog. The worksheet cells used as post map labels can contain math text instructions. In general, the clipboard can be used to cut and paste math text instructions.

Math Text Instruction Syntax

Unless otherwise indicated, all math text instructions begin with a backslash (" \backslash "), and end with a single space. For example, the instruction " $\backslashup50$ " shifts the baseline of the text up 50 percent of the current text height. All characters from the beginning backslash through the ending single space are interpreted as instructions by the math text interpreter, and are not included in the resulting label.

Each line in a text block starts with the default text properties such as typeface, size, color, and style. A line of text within a text block uses the current properties until a math text instruction is encountered. All text following an instruction is modified according to the instruction. For example, if the typeface is changed in the middle of a text string, the text following the instruction will use the new typeface until the end of the line of text is reached, or until another instruction affecting the typeface is encountered.

Math text instructions can also be encapsulated so they are not carried out over an entire line. A left curly brace (" $\{$ ") instructs the math text system to remember all of the text properties in effect at that point. A right curly brace (" $\}$ ") restores the properties to what they were at the matching left curly brace. This allows the insertion of special text in the middle of an otherwise uniform

Appendix B - Math Text Instructions

text block. The only instructions this does not apply to are text baseline instructions (`\dnx` and `\upx`), and the position instructions (`\rpx` and `\spx`). Curly braces can be nested.

To incorporate a backslash, right curly brace, or left curly brace as a text character in a text block, precede them with a backslash when entering the text string. For example, `"\"` produces `"\"`, and `"\"` produces `"{"`.

Instructions based on a percentage, such as font size, are cumulative. This means that a second percentage change within a text block is interpreted as a percentage of the first percentage change. For example, if the font is scaled by 50%, and later in the same text block the font is scaled by 50% again, the font size after the second percentage would be 25% of the original font size.

Math Text Instructions

Instructions that Change Typefaces, Sizes, and Styles

<code>\b</code>	All text after the <code>\b</code> command is bold.
<code>\f"X"</code>	Change to the typeface named X. These are the names listed in the Face drop-down list box in the Text dialog in the plot window. Typeface names are case sensitive. Enclose the face name in double quotes. If the typeface is not found, a generic stick typeface is used in place of the unfound typeface.
<code>\fsX</code>	Change the font size to X% of the current font size. For example, a value of 200 for X increases the font size by two, and a value of 50 for X decreases the font size by one-half.
<code>\i</code>	All text after the <code>\i</code> instruction is italicized.
<code>\plain</code>	This sets the text to normal weight, with no italics, no underlining, and no strikethrough.
<code>\strike</code>	Strikethrough the text.
<code>\ul</code>	All text after the <code>\ul</code> instruction is underlined.

Instructions that Change Text Color

<code>\black</code>	Sets the text color to black.
<code>\blue</code>	Sets the text color to blue.
<code>\green</code>	Sets the text color to green.
<code>\cyan</code>	Sets the text color to cyan.
<code>\red</code>	Sets the text color to red.
<code>\magenta</code>	Sets the text color to magenta.
<code>\yellow</code>	Sets the text color to yellow.
<code>\white</code>	Sets the text color to white.
<code>\gray</code>	Sets the text color to gray.
<code>\rgbbX</code>	Sets the amount of blue in an RGB text color (X = 0 to 255).
<code>\rgbGx</code>	Sets the amount of green in an RGB text color (X= 0 to 255).
<code>\rgbRrX</code>	Sets the amount of red in an RGB text color (X= 0 to 255).

Instructions that Change Text Position

<code>\dnX</code>	Moves the text baseline down X% of the current font size. This instruction produces subscripts or returns the baseline to the original position following a <code>\upX</code> instruction. If a font size (<code>\fsX</code>) instruction follows the <code>\dnX</code> instruction, any subsequent <code>\dnX</code> or <code>\upX</code> instructions are relative to the changed font size.
<code>\rpX</code>	Restores the current position to position #X (X = 1 to 20). This instruction is used in conjunction with the <code>\spX</code> instruction. Any text following this instruction begins at the position defined with the <code>\spX</code> instruction. If the <code>\rpX</code> instruction is used without first setting a position with the <code>\spX</code> instruction, the position for the text is returned to starting position for the text block.
<code>\spX</code>	Saves the current position as position #X (X = 1 to 20). The position is the up, down, left, and right areas within the text block. When the <code>\spX</code> instruction is used, the current location within the text block is assigned a position number. Return to this position using the <code>\rpX</code> instruction. Specify the position number assigned with the <code>\spX</code> instruction when using the <code>\rpX</code> instruction. These instructions are most useful when placing both superscript and subscripts after the same character.
<code>\upX</code>	Moves the text baseline up X% of the current font size. This instruction is used to produce superscripts or to return the baseline to the original position following a <code>\dnX</code> instruction. If a font size (<code>\fsX</code>) instruction follows the <code>\spX</code> instruction, any subsequent <code>\upX</code> or <code>\dnX</code> instructions are relative to the changed font size.

Instructions Used to Insert Special Characters or Date and Time

\aX	<p>Inserts an ANSI character with the code given by X. Generally, this instruction is needed for characters with ANSI numbers beyond the normal limit of the keyboard. For example, use the ANSI character code to include an integral sign in a text string. By determining the character set that includes this symbol and the ANSI code for the symbol, the symbol can be included in the text block. Use the \f"X" instruction to change to the correct character set, followed by the \aX command to specify the correct character number. For example, an integral sign is located at ANSI position 242 in a symbol set called Symbol. Type \f"Symbol" \a242 for the integral sign to appear in the text block.</p> <p>The Character Map program is an accessory program in Windows installations. Use the Character Map to display each of the available character sets with the ANSI code. The ANSI code is displayed in the right half of the Character Map status bar when a character is selected.</p>
\date	<p>Inserts the current date. Be sure to follow this instruction with a space, even if no other text follows the date. The date is updated every time the text is redrawn.</p>
\time	<p>Inserts the current time. The time text is updated whenever the text is redrawn or the graph is printed. Be sure to follow this instruction with a space, even if no other text follows the time.</p>

Examples of Math Text Instructions

Note: Due to page size limitations, some of these examples contain multiple lines of math text instructions. These examples must be entered on one line for the text to be printed correctly.

$\{ \textit{Avogadro} \} \text{ Constant} = 6.022 \times 10^{\{ \text{23} \}}$

$$\textit{Avogadro Constant} = 6.022 \times 10^{23}$$

$\sum_{i=1}^n X_i Y_i = S_{XY}$

$\phi = \phi/x^2 + \phi/y^2$

N_1^5

$\sin^2(X) + \cos^2(X) = 1$

$104^{\circ}37'$

$\frac{a_c}{b_c}$

$\delta^{234}\text{U}$

Appendix C

File Formats

Introduction to File Formats

This appendix contains technical information about the various file formats used by **Surfer**. This information is primarily for advanced users wishing to create their own files or for exchanging information with other applications.

Grid File Formats

There are several **Surfer** grid file formats.

GS ASCII Grid File Format

GS ASCII grid files [.GRD] contain five header lines that provide information about the size and limits of the grid, followed by a list of Z values. The fields within ASCII grid files must be space delimited.

The listing of Z values follows the header information in the file. The Z values are stored in row-major order starting with the minimum Y coordinate. The first Z value in the grid file corresponds to the lower left corner of the map. This can also be thought of as the southwest corner of the map, or, more specifically, the grid node of minimum X and minimum Y. The second Z value is the next adjacent grid node in the same row (the same Y coordinate but the next higher X coordinate). When the maximum X value is reached in the row, the list of Z values continues with the next higher row, until all the rows of Z values have been included.

Appendix C - File Formats

The general format of a GS ASCII grid file is:

id	the four-character identification string DSAA that identifies the file as a GS ASCII grid file
nx ny	nx is the integer number of grid lines along the X axis (columns) ny is the integer number of grid lines along the Y axis (rows)
xlo xhi	xlo is the floating point minimum X value of the grid xhi is the floating point maximum X value of the grid
ylo yhi	ylo is the floating point minimum Y value of the grid yhi is the floating point maximum Y value of the grid
zlo zhi	zlo is the floating point minimum Z value of the grid zhi is the floating point maximum Z value of the grid
grid row 1	these are the rows of floating point Z values of the grid, organized in row order. Each row has a constant Y coordinate. Grid row 1 corresponds to ylo and the last grid row corresponds to yhi. Within each row, the Z values are arranged from xlo to xhi. When an ASCII grid file is created through Surfer , the program includes 10 values per line for readability. When creating an ASCII grid outside of Surfer , you are not limited to a specific number of values per line.
grid row 2	
grid row 3	
...	

The following example grid file is ten rows high by ten columns wide, as indicated by the second row in the header (nx, ny). The first five lines of the file contain header information. X ranges from 0 to 9, Y ranges from 0 to 7, and Z ranges from 25 to 97.19. The first Z value (91.03) corresponds to the lower left corner of the map and the following values correspond to the increasing X positions along the bottom row of the grid file. This file has a total of 100 Z values.

DSAA

10	10			
0.0	9.0			
0.0	7.0			
25.00	97.19			
91.03	77.21	60.55	46.67	52.73
64.05	41.19	54.99	44.30	25.00
96.04	81.10	62.38	48.74	57.50
63.27	48.67	60.81	51.78	33.63
92.10	85.05	65.09	53.01	64.44
65.64	52.53	66.54	59.29	41.33
94.04	85.63	65.56	55.32	73.18
70.88	55.35	76.27	67.20	45.78
97.19	82.00	64.21	61.97	82.99
80.34	58.55	86.28	75.02	48.75
91.36	78.73	64.05	65.60	82.58
81.37	61.16	89.09	81.36	54.87
86.31	77.58	67.71	68.50	73.37
74.84	65.35	95.55	85.92	55.76
80.88	75.56	74.35	72.47	66.93
75.49	86.39	92.10	84.41	55.00
74.77	66.02	70.29	75.16	60.56
65.56	85.07	89.81	74.53	51.69
70.00	54.19	62.27	74.51	55.95
55.42	71.21	74.63	63.14	44.99

Appendix C - File Formats

GS Binary Grid File Format

GS Binary grid files [.GRD] use a layout similar to the ASCII grid format described in the previous section. The only difference is in the identification string and that the files are binary. Data types used in GS Binary grid files include:

Type	Description
char	single byte
short	16 bit signed integer
float	32 bit single precision floating point value
double	64 bit double precision floating point value

The GS Binary format has the following layout:

Element	Type	Description
id	char	4 byte identification string 'DSBB' which identifies the file as a GS Binary grid file
nx	short	number of grid lines along the X axis (columns)
ny	short	number of grid lines along the Y axis (rows)
xlo	double	minimum X value of the grid
xhi	double	maximum X value of the grid
ylo	double	minimum Y value of the grid
yhi	double	maximum Y value of the grid
zlo	double	minimum Z value of the grid
zhi	double	maximum Z value of the grid
z11, z12, ...	float	first row of the grid. Each row has a constant Y coordinate. The first row corresponds to ylo, and the last row corresponds to yhi. Within each row, the Z values are ordered from xlo to xhi.
z21, z22, ...	float	second row of the grid
z31, z32, ...	float	third row of the grid
...	float	all other rows of the grid up to yhi

Surfer Grid and Surfer 7 Grid File Format

Surfer Grid and *Surfer 7* grid files [.GRD] are the same file format. These grids use a tag-based binary format to allow for future enhancements. Each section is preceded by a tag structure, which indicates the type and size of the following data. If a program does not understand or want a particular type of data, it can read the associated tag and quickly skip to the next section. In general, sections can appear in any order except for the first, which must be a *Header* section.

Data types used in *Surfer Grid* and *Surfer 7* grid files:

Type	Description
long	32 bit signed integer
double	64 bit double precision floating point value

Each section is preceded by a tag structure with the following format:

Element	Type	Description
Id	long	The type of data in the following section. See the next table for a list of valid values.
Size	long	The number of bytes in the section (not including this tag). Skipping this many bytes after reading the tag aligns the file pointer on the next tag.

Tag *Id* values. The 0x prefix indicates a hexadecimal value:

Id	Description
0x42525344	<i>Header</i> section - must be the first section within the file.
0x44495247	<i>Grid</i> section - describes a 2D matrix of Z values.
0x41544144	<i>Data</i> section - contains a variable amount of data. The size of the data section is given by the Size field in the tag structure.
0x49544c46	<i>Fault Info</i> section - describes the fault traces used when creating the grid.

The *Header* section must be the first section in the file and has the following format:

Element	Type	Description
Version	long	Version number of the file format. Currently must be set to 1.

Appendix C - File Formats

The *Grid* section consists of a header that describes a 2D matrix of values, followed by the matrix itself. This section encapsulates all of the data that was traditionally referred to as a grid:

Element	Type	Description
nRow	long	number of rows in the grid
nCol	long	number of columns in the grid
xLL	double	X coordinate of the lower left corner of the grid
yLL	double	Y coordinate of the lower left corner of the grid
xSize	double	spacing between adjacent nodes in the X direction (between columns)
ySize	double	spacing between adjacent nodes in the Y direction (between rows)
zMin	double	minimum Z value within the grid
zMax	double	maximum Z value within the grid
Rotation	double	not currently used
BlankValue	double	nodes are blanked if greater or equal to this value

A *Data* section containing the 2D matrix of values (doubles) must immediately follow a *Grid* section. Within the *Data* section, the grid is stored in row-major order, with the lowest row (minimum Y) first

A *Fault Info* section describes the fault geometry used to create the grid. *Fault Info* sections have the following format:

Element	Type	Description
nTraces	long	number of fault traces (polylines)
nVertices	long	total number of vertices in all the traces
data section		variable-sized data block consisting of an array of Trace structures immediately followed by the array of vertices

A *Data* section containing an array of *Trace* structures and an array of *Vertex* structures must immediately follow a *Fault Info* section. The number of *Trace* structures in the array is nTraces, and the number of *Vertex* structures is nVertices.

Trace structure:

Element	Type	Description
iFirst	long	0-based index into the vertex array for the first vertex of this trace
nPts	long	number of vertices in this trace

Vertex structure:

Element	Type	Description
x	double	X coordinate of the vertex
y	double	Y coordinate of the vertex

Example

The following example illustrates the layout for a five row by 10 column grid:

Element	Type	Description
0x42525344	long	Tag: Id for <i>Header</i> section
4	long	Tag: Size of <i>Header</i> section
1	long	<i>Header</i> Section: Version
0x44495247	long	Tag: ID indicating a grid section
72	long	Tag: Length in bytes of the grid section
5	long	Grid Section: nRow
10	long	Grid Section: nCol
0.0	double	Grid Section: xLL
0.0	double	Grid Section: yLL
1.0	double	Grid Section: xSize
1.75	double	Grid Section: ySize
25.0	double	Grid Section: zMin
101.6	double	Grid Section: zMax
0.0	double	Grid Section: Rotation
1.70141e38	double	Grid Section: BlankValue
0x41544144	long	Tag: ID indicating a data section
400	long	Tag: Length in bytes of the data section (5 rows x 10 columns x 8 bytes per double)
Z11, Z12, ...	double	Data Section: First (lowest) row of matrix. 10 doubles
Z21, Z22, ...	double	Data Section: Second row of matrix. 10 doubles
Z31, Z32, ...	double	Data Section: Third row of matrix. 10 doubles
Z41, Z42, ...	double	Data Section: Fourth row of matrix. 10 doubles
Z51, Z52, ...	double	Data Section: Fifth row of matrix. 10 doubles

Level File Format

Level files [.LVL] are used to specify level and property data for various map types within **Surfer**. Level files are ASCII files with one record per line. Fields within the line are separated by commas or white space. Comments are delimited by a single quote character, and may be placed at the beginning of a line or the end of a record.

There are two different forms of the level file format. The first and simplest form consists of a single elevation value per record. For example, the following file can be used to specify logarithmic intervals for a contour map:

```
0.1
1
10
100
1000
```

The second form of the level file format is indicated by specifying the LVL2 keyword on the first line. In this form of level file, each subsequent record has the following general syntax:

Level Flags LineColor LineStyle LineWidth FillForeColor FillBackColor FillPattern FillMode

The fields are space delimited and defined as follows:

Element	Description
Level	elevation of the level
Flags	0=normal, 1=labeled, 2=hached, 3=labeled and hached
LineColor	color of lines used to represent the level
LineStyle	LineStyle is the style of lines used to represent the level. This is one of the line style names as listed in <i>Appendix D, Line Styles, Fill Patterns, and Colors</i> . The name must be enclosed in double quotes.
LineWidth	width of line in mils (thousandths of an inch)
FillForeColor	fill foreground color
FillBackColor	fill background color
FillPattern	FillPattern is one of the fill style names as appears in the fill pattern palette within the dialogs. The name must be enclosed in double quotes.
FillMode	background mode for pattern brushes, 1=transparent, 2=opaque

All colors are specified by name as listed in *Appendix D, Line Styles, Fill Patterns, and Colors* or by the special syntax "Rxxx Gyyy Bzzz." In this form, the red, green, and blue components are substituted for the xxx, yyy, and zzz fields respectively. Each component can range from 0 to 255.

Here is an example of a version 2 level file that defines 4 levels and their properties:

LVL2

```
20 1 "Black" "Solid" 0 "Black" "Black" "Solid" 2
25 0 "Black" "Solid" 0 "R15 G15 B15" "R15 G15 B15" "Solid" 2
30 0 "Black" "Solid" 0 "R31 G31 B31" "R31 G31 B31" "Solid" 2
35 0 "Black" "Solid" 0 "R47 G47 B47" "R47 G47 B47" "Solid" 2
```

Color Spectrum File Format

Color spectrum files [.CLR] are used to define a spectrum, or continuous gradation of colors. This is specified by a series of anchor points with associated colors. The colors between anchor points are interpolated from the nearest anchor points.

The basic format consists of an ASCII file with header information on the first line. Subsequent lines specify anchors points, one anchor point per line.

The header consists of the following space-delimited fields:

Element	Description
Id	case-sensitive string "ColorMap" without the quotes.
Version	format version number. This should be set to 1.
InterpMethod	interpolation method between anchors. This should be set to 1.

Appendix C - File Formats

Subsequent lines define the anchor points, one per line. Each line has the following space-delimited fields:

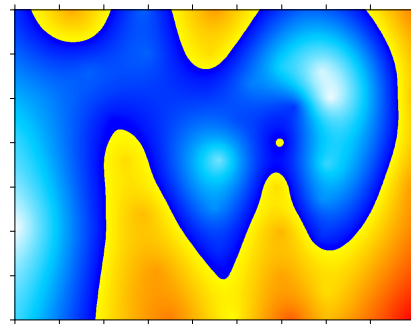
Element	Description
Position	Position is the floating point percentage value (from 0.0 to 100.0). Positions must be specified in increasing order from 0 to 100 percent, and the 0 and 100 percent positions must be specified in the file.
Red	red color component (0 to 255)
Green	green color component (0 to 255)
Blue	blue color component (0 to 255)

In the following example, the anchor points are at 0, 50, and 100. The zero position is blue, the 50 percent anchor is green, and the 100 percent position is yellow.

```
ColorMap 1 1
0 0 0 255
50 0 255 0
100 255 255 0
```

It is also possible to have coincident anchor points in a color file. Anchors and colors are interpreted in order from 0 percent to 100 percent. In the case of coincident points, you can create maps with distinct boundaries, similar to the example shown here.

```
ColorMap 1 1
0 255 0 0
50 255 255 0
50 0 0 255
100 255 255 255
```



In this example, two anchor points lie at the 50 percent position in the [.CLR] file. A distinct change is shown at the 50 percent position on the map.

Import and Export File Formats

Several boundary formats can be imported and exported from **Surfer**. These files can be imported with **File | Import, Map | Base Map**, or used as breakline and fault files. Files are exported with **File | Export**. Exceptions are noted.

There are a number of important distinctions that can be made for different file types because each file format uses a unique set of characteristics. The important file format characteristics are outlined in the following table.

File Format	File Extension	Coordinate Type ²	Drawing Properties ³	Text Blocks ⁴	Content Type ⁵	Import	Export
Bitmap	TIF, BMP, TGA, PCX, DCX, WPG, JPG, GIF, PCT, and PNG	P	X		R	X	X
Golden Software Blanking	BLN	R			V	X	X
Atlas Boundary	BNA	R			V	X	X
Metafile	CGM ¹ , CLP, EMF, WMF	O	X	X	R and V	X	X
USGS Boundary	DLG, LGO, LGS, DDF	R			V	X	
AutoCAD DXF Drawing	DXF	RO	some	some	V	X	X
ESRI ArcInfo Export	E00	R		X	V	X	
Golden Software Boundary	GSB	R			V	X	X
Golden Software Interchange	GSI	R	X	X	R and V	X	X
MapInfo Interchange	MIF	R	some	X	V	X	X
Golden Software PlotCall	PLT	O		X	V	X	
ESRI Shapefile	SHP	R			V	X	X

¹ CGM is export only

² R = real world coordinates such as latitude/longitude, UTM, etc.

P = pixel coordinates

O = other (page coordinates, metafile coordinates, etc.)

³ Contains properties such as line color, fill, etc.

⁴ Text blocks are editable. (When the *All text as lines* or *All text as areas* options are chosen as export options, the text is not editable.)

⁵ Files can contain raster (bitmap) information, vector (lines) information, or both. There is no raster to vector conversion in **Surfer**. If you have a bitmap on screen and try to export it as a vector format, such as a DXF, the resulting file will be


Appendix C - File Formats

blank. If you have a bitmap on the screen and have drawn polylines or polygons and export as a vector format, only the drawing objects are included in the file. USGS SDTS DDF boundary files contain vector information (Topological Vector Profile). The DEM version of this file is considered a raster file (Raster Profile).

V = contains vector information only

R = contains raster information only

R and V = contains raster and vector information

In addition to the chart on the previous page, the file formats used in **Surfer** are discussed briefly below. For up-to-date information on the import and export options, click the help button  in the file type import or export dialog.

Golden Software Boundary Files

Golden Software Boundary [.GSB] files contain geographic data in a binary format. In **Surfer**, this format includes polylines, polygons, and points. The [.GSB] format uses coordinates from the *Export Rectangle*. This format is a spatial format, so line properties, fill properties, symbol properties, and text are not retained in the file. If you want to save your work as it exists on the screen, you must use the **File | Save** or **File | Save As** commands to save in the [.SRF] format.

Golden Software Blanking Files

Blanking files [.BLN] are used to represent boundary information, faults, breaklines, and for blanking polygonal regions within grids. Blanking files are comma-delimited ASCII files consisting of a repeating series of polygon or polyline definitions. Each definition has the following format:

Blanking files are comma-delimited ASCII files consisting of a repeating series of polygon or polyline definitions. Each definition has the following format:

npts, flag	This is the header for the first boundary in the blanking file. <i>npts</i> is the number of XY coordinate pairs used to define this boundary. <i>flag</i> defines the type of blanking. If you are using this boundary as a blanking file, a value of 0 is used to blank outside the defined boundary, a value of 1 blanks inside the defined boundary.
x1, y1	first coordinate pair of the boundary
x2, y2	second coordinate pair
...	coordinate pairs
xn, yn	This is the last coordinate pair of the boundary. For a polygon, this point must be identical to the first coordinate pair.

The sequence may be repeated any number of times within the same file to define multiple polygons and polylines. When using multiple polylines in a blanking operation, the *flag* value in the polygon header should be 1 for all boundaries; otherwise, the entire grid is blanked.

The following example file defines a polygon that blanks all grid nodes falling within the shape. Notice that the header contains the number of XY pairs in the file (8), and the flag value is "1," specifying blanking inside the rectangle.

```
8, 1
84.6601, 34.0783
84.739, 34.0797
84.9233, 34.0829
85.0474, 34.0969
85.006, 34.4034
84.9547, 34.3848
84.654, 34.4127
84.6601, 34.0783
```

The following example defines two breaklines for input into the gridding module:

```
2, 1
3.0, 4.5, 1.0
0.5, 4.5, 2.0
3, 1
4.5, 0.0, 1.1
8.5, 0.0, 1.2
8.5, 1.5, 1.4
```

Golden Software PlotCall Files

Golden Software PlotCall [.PLT] files are produced with the DOS based program PlotCall. PlotCall was included with the DOS versions of **Surfer** and **Grapher**. The [.PLT] import option allows you to import these files for use in **Surfer 8**.

Golden Software Interchange Files

Golden Software Interchange [.GSI] files are binary files containing records indicating graphical entities and their properties. These include boundary objects, coordinates, color, fill patterns, and line styles. These files provide a medium of exchange between Golden Software application programs. We are in the process of adding this file format to our other applications, so the [.GSI] format is not necessarily available in some versions of Golden Software applications.

USGS SDTS Topological Vector Profile Files

The SDTS file is a newer USGS file format. There are two separate types of SDTS files. The Topological Vector Profile SDTS files contain boundary line information and the Raster Profile SDTS files contain gridded elevation information. The Raster Profiles are used for creating grid-based maps such as contour maps. Topological Vector Profiles are used as base maps (**Map | Base Map**). An uncompressed SDTS file contains several files with the [.DDF] extension. You can choose any one of these files in the **Import File** dialog. You must have all of the [.DDF] files in a single folder to import a map. For example, you cannot copy just one [.DDF] file into a new folder and create a map from it.

USGS Digital Line Graph Files

USGS Digital Line Graph files contain digital geographic data for the entire U.S. produced by the U.S. Geological Survey. The data is at 1:24,000, 1:100,000, or 1:2,000,000 scale. These files can contain polygons, polylines, or points representing several different types of boundaries.

The Digital Line Graph formats that **Surfer** can read are

- USGS Digital Line Graph [.DLG]
- USGS Digital Line Graph (Optional) [.LGO]
- USGS Digital Line Graph (Standard) [.LGS]

Bitmaps

Bitmaps save the file in a format recognized by many applications. Bitmaps contain a pixel image of the current plot window contents, including the polylines, polygons, points, and labels as they are seen on the screen.

Because bitmap files are raster images, they cannot be scaled in the receiving application without distortion or some loss of resolution. They can be useful for reproducing the file in other applications.

If you need to reproduce and rescale the file in another application, you might consider using a vector format.

Surfer does not support georeferenced bitmaps. The bitmap can be imported into **Surfer**, but it will contain pixel coordinates rather than real-world coordinates.

Surfer currently supports the following bitmap (raster) file formats:

Tagged Image Format (TIFF) [.TIF]	WordPerfect Bitmap [.WPG]
Windows Bitmap [.BMP]	DCX (Multi-page PCX) [.DCX]
Targa (TrueVision) [.TGA]	JPEG Compressed Bitmap [.JPG]
PCX (ZSoft/Paintbrush) [.PCX]	PICT (Macintosh) Bitmap [.PCT]
CompuServe Bitmap [.GIF]	Portable Network Graphics Bitmap [.PNG]

Metafiles

Windows metafiles contain drawing information for reproducing vector images. Metafiles conserve the symbol, line style, fill, and text information of objects as Windows drawing commands. Because of this, Windows metafiles can be rescaled in the receiving application without any distortion or loss of resolution. Metafiles use a coordinate system based on screen coordinates, and do not contain true "map" or real-world coordinates.

The Windows metafile formats are Windows Clipboard (Picture) [.CLP], Windows Metafile (Enhanced) [.EMF], Windows Metafile [.WMF], and Computer Graphics Metafile [.CGM] (export only).

Appendix C - File Formats

Atlas Boundary Files

Atlas Boundary [.BNA] files are ASCII format files that contain polygons, polylines, points, and coordinates. The Atlas Boundary format is a spatial format, so line properties, fill properties, symbol properties, and text are not retained in the file. Because these are ASCII format files, you can open them in programs that can read ASCII text files (including most spreadsheet programs and word processing programs), so you can view and edit the actual file contents.

The general format for an Atlas Boundary file is:

Pname 1, Sname 1, type/length

x1, y1

x2, y2

x3, y3

...

xn, yn

Pname 2, Sname 2, type/length

x1, y1

x2, y2

...

xn, yn

An alternate format for Atlas Boundary files is:

Pname 1, Sname 1, type/length, x1, y1, x2, y2, x3, y3, ... xn, yn

Pname 2, Sname 2, type/length, x1, y1, x2, y2, x3, y3, ... xn, yn

Element	Description
Pname	Primary ID name enclosed in double quotes (not used in Surfer)
Sname	Secondary ID name enclosed in double quotes (not used in Surfer)
type/length	the type of entity to follow and the number of vertices
x, y	X and Y coordinates of the vertices making up the entity. The number of vertices is controlled by the <i>type/length</i> field (see next table)

The type/length field is interpreted as follows:

type/length	Description
< -1	negative integers less than -1 indicate a polyline, the number of vertices to follow is equal to the absolute value of type/length
1	point entity, one vertex follows
2	2 indicates an ellipse. Two "vertices" follow. The first vertex is the center of the ellipse. The second "vertex" contains the major and minor radii respectively. If the minor radius is 0, the ellipse is a circle.
> 3	values greater than 3 define a closed polygon, the value indicates the number of vertices to follow

Multiple entities may be concatenated one after another within the same file.

Two kinds of polygons may be specified: simple and compound. A simple polygon contains a starting point, a series of points specifying the polygon's boundary and a closing point with the same coordinate as the starting point. A compound polygon contains one or more sub-polygons, such as islands or lakes. Atlas Boundary files use a special technique to specify the sub-polygons comprising compound polygons.

Assume a compound polygon that looks like a torus (donut). Assume the outer boundary is sub-polygon A, and the inner excluded region (the donut hole) is sub-polygon B. Here is how the coordinates are specified:

AX1, AY1	Starting point of sub-polygon A
AX2, AY2	Points specifying boundary of sub-polygon A
.	.
AXn, AYn	Ending point of sub-polygon A
BX1, BY1	Starting point of sub-polygon B
BX2, BY2	Points specifying boundary of sub-polygon B
.	.
BXn, BYn	Ending point of sub-polygon B
AX1, AY1	Return path to start of sub-polygon A (flag point)

The last vertex in the example above deserves special mention. Each sub-polygon after the first must contain a "return path" to the first vertex of the first sub-polygon. This vertex is used as a marker (called a flag point) to indicate the end of each sub-polygon. This means the first sub-polygon vertex must be unique and cannot appear as a coordinate within any other sub-polygon.

ESRI ArcInfo Export Format Files

ESRI ArcInfo Export Format [.E00] files are ASCII files containing topological entities and their properties. They provide a medium of exchange between ESRI application programs on different hardware and operating system platforms (Windows, UNIX, etc.).

Disclaimer: The [.E00] file format is not publicly documented by ESRI. Although Golden Software has tested this [.E00] import filter software with a number of publicly available [.E00] files, it may not be compatible with all [.E00] files created by all versions of ESRI application programs. Golden Software is not affiliated with ESRI, and this import filter software is not a product of, nor endorsed by, ESRI.

ESRI Shapefiles

ESRI Shapefiles [.SHP] contain boundary objects including polygons, polylines, or points. The objects do not have associated properties such as color or line style. A shapefile may contain polyline objects, polygon objects, or point objects, but not a combination of different object types.

Each [.SHP] file is usually accompanied by files with the same name, but with the [.SHX] and [.DBF] extensions. These files are not used in **Surfer**.

MapInfo Interchange Format Files

MapInfo Interchange Format [.MIF] files contain boundary objects including polygons, polylines, and points. The objects optionally have properties, such as color or line style, associated with them.

Each [.MIF] file is usually accompanied by a file with the same name, but with the [.MID] extension. The [.MID] file contains attribute information about the objects. This information is not used in **Surfer**.

AutoCAD DXF Files

All the boundary objects are preserved in the AutoCAD DXF format, including polygons, polylines, points, rectangles, circles, text, and so on. Some colors are contained in the drawing and in some cases, you can preserve color fill upon export.

Data File Formats

Surfer supports several data file formats including ASCII data files and spreadsheet files. The data file formats are discussed briefly below.

ASCII Data File Formats

There are three common ASCII data formats [.DAT], [.CSV], and [.TXT]. ASCII files are generic format files read or produced by most applications. These files can also be imported into most applications, including word processors, spreadsheets, and ASCII editors.

ASCII files do not contain any worksheet formatting information such as row height, column width, or cell formatting. When ASCII files are loaded into the worksheet, the default column formatting parameters are applied to the data. This does not result in any change to data, but might result in rounding of values in the data display. There is no limitation on the number of rows or columns in an ASCII format. ASCII formats save and load slowly because of the conversion from binary numbers to character representation.

There are some distinctions in formatting of ASCII files. Here are some brief notes that outline the usefulness of the ASCII file features.

- *Delimiters* control the separation between cell entries in a file. Spaces, tabs, semi-colons, or commas can be used to separate cells. If cell entries contain spaces in text, the comma or semi-colon delimiters are necessary if quotes are not used to qualify the text. Otherwise, the text string would be interpreted as two cell entries rather than a single entry.
- *Placing Quotes around Text* - There are two types of entries in an ASCII file, values and text. Values are actual numbers, while text is any type of character, including numbers, symbols, and text characters. Single or double quotes can be placed around text strings. If a number should be interpreted as text, surround it with double quotes. When text strings contain spaces, it is recommended to use single or double quotes around text cell entries.
- *Using Commas or Semicolons in Addition to Quotes* - Although double quotes are not mandatory around text strings, they are useful when creating a space-delimited file that contains text. Often there are text strings that contain spaces, as in a date containing month name, day and year. With space delimited files this single entry is interpreted as more than one cell when loading this file into the worksheet. The safest way to eliminate this problem is to place double quotes around all text strings and use comma delimiting between variables.

When saving [.DAT] files, you are prompted to choose the delimiters and how to define text strings. [.TXT] files are tab delimited ASCII text files with no quotes around the text strings. [.CSV] (comma separated variables) files are comma delimited with double-quotes around text strings (non-numeric or mixed alpha numeric).

Appendix C - File Formats

Excel Data Files

Microsoft Excel [.XLS] files contain data and retain some cell formatting in **Surfer**. Other types of information, such as formulas, are ignored.

Excel files can preserve all formatting information available in the Golden Software worksheet. This format has a 65,536-row limit and a 256-column limit. Therefore, this format cannot be used to store very large data sets.

Special Characters Used in Excel Files

There are a number of special characters that can be contained in an Excel file that the worksheet cannot handle in the same way as Excel. For these characters, **Surfer** substitutes a reasonable representation so the value displayed in the cell will look similar to what was displayed in Excel. See the online help topic, *Special XLS Characters*, for more information.

SYLK Files

Microsoft SYLK [.SLK] files contain data and worksheet formatting information except for background color. When SYLK files are imported into the worksheet, the data format is based on the information in the file. If the data has fixed formatting with four decimal digits the data are displayed in this manner when the [.SLK] file is loaded into the worksheet.

If a SYLK file is created in another application and loaded into the worksheet, there might be special formatting information in the file that the worksheet cannot use. In these cases, the data file is loaded without the special formatting. If the file is saved in an SYLK format in the worksheet the special formatting information is lost.

Technically, the SYLK specification has a 65,536 row limit and a 256 column limit. However, SYLK is an ASCII format so it is possible to have an unlimited size. The Golden Software SYLK Import/Export filters allow limits greater than the specifications indicate so larger worksheets can be saved in SYLK format. However, other spreadsheet programs may not be able to load these large SYLK files.

Lotus Data Files

These are files produced from Lotus programs including 1-2-3 and Symphony. The worksheet can import these files but cannot save in any Lotus format. [.WKS], [.WK1], [.WK2], [.WK3], [.WK4], [.WR1], and [.WRK] files can be read into the worksheet. The Lotus import filter does not import formatting information or formulas. Also, the newest Lotus file formats cannot be directly imported into the worksheet, they must be saved in one of the recognized formats.

Appendix D

Line Styles, Fill Patterns, and Colors

Introduction to Line Styles, Fill Patterns, and Colors

The line style, fill pattern, and color names are used for certain advanced operations:

- Some settings in the **File | Preferences | Dialog Defaults** require that the names be entered in the format found in the line, pattern, and color palettes.
- Creating a level file manually requires the line style names and color names to be entered as they appear in the respective palettes.
- Automating **Surfer** requires that line style names and pattern names are entered as they appear in the dialog.

When using the names in default settings, level files, and in automation, enter the name exactly as it appears below. The names are not case sensitive, but they must include any spaces and punctuation as shown below. If you create custom colors, lines, or patterns, use the name you assign in the custom property dialog.

Line Styles

Invisible	.4 in. Dash	Dash Dash Dot
Solid	.5 in. Dash	Dash Dash Dot Dot
.1 in. Dash	Dash Dot	Dash Dash Dot Dot Dot
.2 in. Dash	Dash Dot Dot	
.3 in. Dash	Dash Dot Dot Dot	

Appendix D - Line Styles, Fill Patterns, and Colors

Fill Patterns

There are three types of fill patterns: stock, bitmap, and vector. Vector and stock patterns can have transparent backgrounds. Vector patterns are scalable. The **File | Preferences | General Simulate stock hatch patterns** option only applies to stock patterns.

None	75.0% Black	Igneous
Solid	87.5% Black	Schist
Diagonal Cross	Sand	Gravel
Crosshatch	Swamp	Iron
Forward Slash	Forest	Spheres
Backward Slash	Orchard	Solid Dash
Vertical	Water	Thatch
Horizontal	Limestone	Checkerboard
6.25% Black	Tilted Limestone	Shingle
12.5% Black	Dolomite	Dolomite2
25.0% Black	Shale	
50.0% Black	Granite	

Colors

Automation does not require color names because there are explicit enumerations for the colors. Refer to the *SrfColor Values* or *WksColor Values* topics in online help for a list of color enumerations. Colors can also be referenced by RGB values. The syntax is "Rxxx Gyyy Bzzz" where xxx, yyy, and zzz specify a red, green, and blue color component respectively. Each color component can range from 0 to 255.

For example:

"R0 G0 B0" All components are 0, resulting in black
"R0 G255 B0" Pure green
"R255 G255 B255" All components are at full intensity, resulting in white.

Color names:

Black	80% Black	60% Black
90% Black	70% Black	50% Black

Appendix D - Line Styles, Fill Patterns, and Colors

40% Black	Kentucky Green	Ruby Red
30% Black	Light Green	Brick Red
20% Black	Spring Green	Tropical Pink
10% Black	Turquoise	Soft Pink
White	Sea Green	Faded Pink
Blue	Faded Green	Crimson
Cyan	Ghost Green	Regal Red
Green	Mint Green	Deep Rose
Yellow	Army Green	Neon Red
Red	Avocado Green	Deep Pink
Magenta	Martian Green	Hot Pink
Purple	Dull Green	Dusty Rose
Orange	Chartreuse	Plum
Pink	Moon Green	Deep Violet
Dark Brown	Murky Green	Light Violet
Powder Blue	Olive Drab	Violet
Pastel Blue	Khaki	Dusty Plum
Baby Blue	Olive	Pale Purple
Electric Blue	Banana Yellow	Majestic Purple
Twilight Blue	Light Yellow	Neon Purple
Navy Blue	Chalk	Light Purple
Deep Navy Blue	Pale Yellow	Twilight Violet
Desert Blue	Brown	Easter Purple
Sky Blue	Red Brown	Deep Purple
Ice Blue	Gold	Grape
Light BlueGreen	Autumn Orange	Blue Violet
Ocean Green	Light Orange	Blue Purple
Moss Green	Peach	Deep River
Dark Green	Deep Yellow	Deep Azure
Forest Green	Sand	Storm Blue
Grass Green	Walnut	Deep Blue

Index

- # cols · 387
- # of lines · 95, 98, 99
- # of nodes · 400
- # rows · 387
- %uses statement · 525, 540
- [.BLN] · 65, 86, 232, 239, 240, 454, 455, 466, 586–87
- [.BMP] · 232, 589
- [.BNA] · 65, 86, 232, 466, 590–91
- [.CGM] · 589
- [.CLP] · 232, 589
- [.CLR] · 51, 260, 267, 268, 363–70, 583
- [.CSV] · 65, 82, 593
- [.DAT] · 22, 65, 82, 239, 409, 454, 593
- [.DBF] · 592
- [.DCX] · 232, 589
- [.DDF] · 89, 160, 197, 232, 588
- [.DEM] · 89, 160, 197
- [.DLG] · 232
- [.DT*] · 89, 197
- [.DT0] · 161
- [.DT1] · 161
- [.DT2] · 161
- [.DXF] · 37, 230, 232, 466
- [.E00] · 232, 592
- [.EMF] · 232, 589
- [.GIF] · 232, 466, 589
- [.GRD] · 24, 160, 197, 409, 575–81
- [.GSB] · 232, 466, 586
- [.GSI] · 232, 588
- [.HDR] · 89, 160, 197
- [.JPG] · 232, 464, 466, 589
- [.LGO] · 232
- [.LGS] · 232
- [.LVL] · 203, 299, 301, 370–71, 582
- [.MID] · 592
- [.MIF] · 232, 592
- [.PCT] · 232, 466, 589
- [.PCX] · 232, 589
- [.PLT] · 232, 587
- [.PNG] · 232, 589
- [.SET] · 552, 559
- [.SHP] · 232, 592
- [.SHX] · 592
- [.SLK] · 65, 594
- [.SRF] · 6, 36, 463, 466
- [.STX] · 160
- [.TGA] · 232, 589
- [.TIF] · 232, 466, 589
- [.TXT] · 65, 82, 593
- [.W??] · 594
- [.WMF] · 232, 466, 589
- [.WPG] · 232, 589
- [.XLS] · 65, 81, 594
- <&> · 85
- <C> · 85
- <D> · 85
- <F> · 85
- <P> · 85
- <T> · 85
- 1.70141e+038 · 403
- 16 colors · 464
- 1-grid vector map · 273–91. *See also* vector map
 - creating · 279
- 256 colors · 464
- 25th percentile · 73
- 2-grid vector map · 273–91. *See also* vector map
 - creating · 279
 - data · 275–78
- 3D DXF · 230
 - contour lines · 37
- 3D label lines · 251

Index

75th percentile · 73
95% or 99% confidence interval for the mean
· 74

A

about Surfer · 16
absolute value · 359
 label format · 480
accessing objects · 494
accumulated horizontal distance · 454
across and then down · 85
active cell · 21, 57, 58
active cell edit box · 21, 57, 58
active cell location · 57, 58
ActiveDocument · 503
ActiveWindow · 503
ActiveX automation members · 495, 533
ActiveX automation objects · 494
ActiveX object · 527
add contour labels · 225
add grid · 399
add label · 32
add levels · 203, 206
add method · 501, 503
add watch · 538
adding anchor nodes · 365
addition · 561
adjacent maps · 376
 combining (overlay) · 378
affected levels
 contour labels · 221
 fill · 214
 hachures · 227
 line properties · 209, 211
align maps · 375
align objects · 375, 484, 488
 snap to ruler · 375
alignment
 text · 468
 worksheet · 67
all · 93
all sessions · 553
600

alpha · 392
always reset · 553
always split · 509
ambient · 322
anchor node
 adding · 365
 colors · 366
 deleting · 366
 positioning · 366
and · 94, 564
angle · 138, 247, 250
 anisotropy · 109, 187
 axis tick labels · 357
 axis title · 352
 classed post map · 249
 search ellipse · 106
 symbol · 245
 variogram · 185
 vector map · 282
angle (direction) data
 vector map · 277
angle grid · 278
angular divisions · 173
anisotropically rescaled relative separation
 distance · 185
anisotropy · 108–10, 164, 186
 angle · 185, 187
 example · 110
 ratio · 185, 187
aperture · 549
application · 465
application object · 494, 501, 502, 504, 516
application property · 501
apply zones to lines of constant · 300
approximate density · 145
areas · 445, 448–50
 blanked planar · 445
 negative planar · 445, 448
 negative surface · 445, 449
 positive planar · 445, 448
 positive surface · 445, 449
 total planar · 445
arguments · 520

- arithmetic average · 72, 143
- arrange
 - align objects · 375, 484, 488
 - break apart · 461, 483
 - combine · 483
 - free rotate · 483, 488
 - move backward · 482
 - move forward · 482
 - move to back · 482
 - move to front · 482
 - order objects · 477, 482
 - order overlay · 482
 - rotate · 483, 488
 - transform · 483, 487, 488
- arrange maps · 373
- arrange overlays · 377
- arranging objects
 - object manager · 492
- array · 158
- array of Z values · 158
- array variables · 516
- arrow direction
 - vector map · 274
- arrow length
 - vector maps · 274
- ascending sort · 68
- ASCII data files · 454, 593
- ASCII delimiters · 593
- ASCII file formatting · 82
- ASCII format file · 466
- ASCII grid · 409
- ASCII grid file format · 575–81
 - example · 577
- ASCII text file · 443
- ASCII XYZ · 409
- assign object ID · 485
- Atlas boundary · 590–91
- Atlas boundary file · 232, 466
 - spatial format · 466
- AutoCAD DXF · 230, 466, 592
 - spatial file format · 466
- AutoCAD DXF drawing file · 232
- autocorrelation · 119
 - variograms · 176
- autocorrelation estimator · 178
- autocovariance
 - variograms · 176
- autocovariance estimator · 177
- autofit · 188–90
 - least absolute value · 189
 - least squares · 189
 - max lag distance · 190
 - maximum iterations · 189
 - parameter fit limits · 190
 - restarting · 189
 - target precision · 189
- automatic compression · 464
- automatic scaling · 350
- automatic variogram fitting dialog · 188
- automating Surfer · 493–546
- automation
 - examples · 540–46
 - property names · 595
- automation help · 495, 499
- automation objects · 502, 494–508
- average · 93, 143, 564
- average deviation · 76
- average distance · 145
- averages
 - weighted · 157
- axes · 33–36, 347–61, 376
- axes collection · 495, 508, 546
- Axes collection · 500
- axis · 200, 242, 260, 268, 274, 295, 316, 347–61
 - automatic scaling · 350
 - bottom · 347
 - display · 349
 - grid lines · 360
 - left · 347
 - length · 39
 - limits · 350
 - maximum · 350
 - minimum · 350
 - position · 349, 351
 - properties · 33–36

Index

- right · 347
 - scaling · 35, 349
 - select · 33, 348
 - tick spacing · 350
 - ticks · 35
 - title · 34
 - top · 347
 - vector map · 289
 - X · 347
 - Y · 347
 - axis attributes · 360
 - axis length · *See* map scale
 - axis line properties · 360
 - axis object · 496, 508, 546
 - axis plane
 - axis tick labels · 358
 - axis properties · 347–61
 - labels · 35
 - tick spacing · 35
 - axis scaling
 - limits · 343
 - axis tick labels
 - angle · 357
 - axis plane · 358
 - decimal places · 359
 - label format · 357, 359
 - offset · 357
 - show · 357
 - axis ticks · 353
 - editing · 355
 - spacing · 355
 - axis title · 352
 - angle · 352
 - font · 352
 - offset along axis · 352
 - offset from axis · 352
 - background example · 346
 - background mode · 213, 555
 - backup files · 547
 - base · 294, 297, 311, 316
 - surface · 325, 327
 - base map · 8, 231–36, 237, 238, 461, 585
 - coordinates · 237
 - creating · 232
 - file formats · 232
 - fill properties · 233
 - font · 235
 - importing · 461
 - limits · 231, 237, 343
 - line properties · 233
 - overlays · 237
 - polygons · 235
 - polygons and polylines · 233
 - polylines · 235
 - properties · 233
 - scale · 237
 - symbol properties · 233
 - symbols · 235
 - base map file formats · 232
 - base map page · 234
 - base object · 500
 - base properties
 - surface · 325
 - based on font · 256
 - basic language · 509
 - BASIC language · 513, 534
 - BASIC language help · 511
 - basis function · 132
 - bessel functions · 562
 - between cols · 385
 - between rows · 385
 - biharmonic operator · 435
 - bilinear interpolation · 262
 - bilinear interpolation resampling · 399, 401
 - bi-linear saddle · 131
 - binary grid · 409, 578
 - binning method · 252
 - bins · 241
 - bitmap · 259, 462
-
- B**
- backcolor · 555
 - background · 200, 242, 260, 268, 274, 295, 316, 331, 345–46
 - background color · 67, 555
- 602

JPEG compression · 464
bitmap coordinates · 236
bitmap export · 463
bitmap size · 463
bitmaps · 588
blank · 216, 403–5
blank edge of grid · 396
blank node
 grid node editor · 407
blank node if more than this many sectors
 are empty · 105
blanked areas
 color fill · 403
 contour lines · 403
 contour map · 216
 grid data · 404
 math calculations · 403
 wireframes · 403
blanked grids
 volume · 453
blanked nodes · 398
 surface · 318
blanked planar area · 445
blanked regions · 202, 216
 cut and fill volume · 446
 in volume calculations · 443
 shaded relief map · 267
 volume · 448
 wireframe · 313
blanking · 403–5
 individual grid nodes · 405
blanking file · 240
 breaklines and faults · 111
 cross section · 455
blanking file format · 586–87
 example · 587
blanking inside or outside a polygon · 404
blanking outside multiple polygons · 405
blanking value
 1.70141e+038 · 403
blanking values
 gridding search · 106
blend colors · 363, 365

BLN · 65, 86, 232, 239, 240, 454, 455, 466,
 585, 586–87
block kriging · 117, 119
block select · 476
BMP · 232, 585, 589
BNA · 65, 86, 232, 466, 585, 590–91
bold · 570
bold text · 12
Boolean · 515
Boolean expressions · 564
 grid data · 94
 grid function · 94
 grid math · 94
 grid variogram · 94
Boolean functions
 filtering data · 94
border · 298
bottom axis · 347
boundary files · 6
boundary map · *See* base map
boundary number · 454
boundary tension · 123, 124
bounding box · 477
break apart · 461, 483
break apart metafiles · 462
break apart overlay · 380
break bar · 510
breakline · 112, 111–13
 blanking file · 111
breakpoint · 539
browse · 533
 debug · 495
by grid file · 283
by magnitude · 283
byte · 515

C

CAD · 466
calculate from input extents · 400
calculations · 561
calculus · 418–41
 biharmonic operator · 435

Index

- correlogram · 439
- differential and integral operators · 434, 433–36
- directional derivatives · 421–26
- equation basis · 419
- first derivative · 423
- Fourier and spectral analysis · 437–41
- integrated volume · 435
- Laplacian operator · 434
- periodogram · 439
- profile curvature · 430
- references · 442
- second derivative · 424
- tangential curvature · 432
- terrain aspect · 429
- terrain modeling · 427–33
- terrain slope · 428
- canceling gridding · 98
- Cartesian coordinates
 - vector map · 281
- Cartesian data · 273, 275
- cell alignment · 67
- cell background color · 67
- cell formatting · 66
 - save · 64
- cell properties
 - alignment · 67
 - number · 66
- cells · 58, 505
 - deselecting · 62
 - selecting · 62
 - selecting blocks · 62
- center alignment
 - worksheet · 67
- center on page · 84
- centimeters · 550
- central difference · 271
- central weight · 392
- CGM · 585, 589
- change colors · 50
- change data file · 247
 - classed post and post maps · 253
- change grid file · 201, 261, 269, 280, 281, 318
- changing drawing order · 482
- chart
 - object model · 496
- check for update · 14
- choose objects · 475–77
- circle · 474
- class · 254
- class module · 511, 525, 526
 - example · 527
- classed post map · 8, 241
 - 3D label lines · 251
 - binning method · 252
 - change data file · 253
 - classes page · 252
 - create · 246
 - data files · 243
 - general · 254
 - general page · 249
 - labels · 250–51
 - legend · 254–56
 - properties · 247–53
 - symbol · 253
 - symbol angle · 249
 - symbol frequency · 249
 - updating data · 253
 - Z values · 244
- classed post maps
 - components · 242
- classes · 241
- classes page · 252
- classical variogram estimator · 177
- clear · 63
- clear transforms · 483, 489
- click · 475
- clip symbols at map limits · 282, 288
- close
 - code sheet · 509
- close all · 511
- close method · 501
- CLP · 232, 589
- CLR · 51, 260, 267, 268, 363–70, 583

- code editor window · 509
- code module · 511, 525
- code sheet tabs · 509
- coefficient of kurtosis · 79
- coefficient of skewness · 78
- coefficient of variation · 77, 143
- coefficient of variation filter · 394
- collapse object manager list · 491
- collection
 - Axes · 500
 - Documents · 500
 - Levels · 500
 - Overlays · 500
 - retrieving items · 500
 - ScaleBars · 500
 - Selection · 500
 - Shapes · 500
 - Windows · 500
- collection objects · 495, 500
- color · 50, 363–71, 555
 - anchor node · 366
 - blending · 365
 - custom · 481
 - fill · 479
 - font · 556
 - line · 479
 - names · 595, 596
 - RGB · 554
 - Scripter keywords · 509
 - surface · 316
 - symbol · 479, 557
 - text · 479, 571
 - vector maps · 274
- color 3D surface · 47
- color depth · 464
- color fill · 30, 199, 363
 - contour · 212–17
 - contour map · 201
- color modulation · 324
- color palette · 365
- color scale bar · 260, 263, 316
 - add · 218, 264, 286, 309, 328
 - blanked areas · 216
 - contour map · 218–19
 - edit · 218, 265, 286, 310, 329
 - image map · 264–65
 - removing · 219, 264, 287, 309, 328
 - surface · 328–29
 - vector map · 274, 286–87
 - visibility · 219, 264, 287, 309, 328
 - wireframe · 301, 309–10
- color scaling · 365
 - vector map · 283
- color spectrum · 31, 51, 212, 363–71
 - add anchor node · 365
 - data color to mapping · 370
 - data to color mapping · 366
 - delete anchor nodes · 366
 - load · 366
 - minimum and maximum values · 368
 - opening · 364–65
 - position anchor node · 366
 - reusing · 367
 - save · 367
 - slider button · 365
 - use data limits · 366
 - value · 365
- color spectrum dialog · 365–67
- color spectrum file · 363–70
 - creating · 369
 - using · 370
- color spectrum file format · 583
 - example · 584
- color spectrum files · 260, 268
- color spectrums
 - in map series · 367
- color vectors · 365
- color zones · 40, 294, 306–9
 - adding and deleting · 309
- color zones page · 300
- colors
 - default settings · 554
 - image map · 263
 - shaded relief map · 271
- column
 - headings · 54

Index

- column letters · 57
- column width · 64
- columns · 505
 - data files · 53
- combine maps · 44. *See* overlays
- combine objects · 483
- combine Surfer files · 375
- comma delimited · 82
- comma separated variables · 593
- command prompt
 - running scripts from · 536
- comments · 514
- common map properties · 331
- compact · 359
 - label format · 480
- compare maps · 367
- comparison operators · 94, 564
- compass gradient filters · 395
- composite · 461
- composite maps · *See* overlays
- composite object · 461
- CompuServe bitmap · 232, 589
- computer graphics metafile · 589
- confidence interval · 74
- contact information · 16
- context-sensitive help · 15
- continue
 - line · 514
- contour label
 - editing · 32
- contour labels · 199, 220–25
 - add · 225
 - cancel edits · 225
 - delete · 225
 - edit position · 224
 - frequency basis · 221
 - move · 224
 - regular interval · 220
 - remove all · 224
 - spacing · 223
 - specific levels · 221
- contour level
 - change properties · 28
- contour levels · 25, 199, 204, 205, 204–6, 304. *See Also* levels
 - affected levels · 209
 - frequency basis · 209
 - gradational color fill · 213
 - grid node editor · 407
 - hachures · 227, 225–28
 - line properties · 210
 - regular interval · 205
 - uniform line properties · 209
- contour line properties · 199
- contour lines · 198, 203, 207–11
 - "wrong" relative to data · 150
 - color · 207
 - export · 230
 - gradational · 207
 - index · 211
 - invisible · 217
 - regular interval · 205
 - smoothing · 229
- contour map · 7, 24–32, 363
 - 3D DXF export · 37
 - add label · 32
 - color fill · 30
 - color scale bar · 218–19
 - components · 199
 - creating · 25, 200
 - creating in Scriptor · 540
 - defined · 197
 - delete label · 32
 - drawing contours · 198
 - edit labels · 32
 - export contours · 37
 - fill · 199, 201, 363, 364
 - general page · 201
 - hachures · 199, 225–28
 - interval · 27
 - level of detail · 204
 - level properties · 28
 - levels · 25, 199
 - levels page · 203
 - line properties · 29
 - move label · 32

- properties · 25, 200, 201–6
- smoothing · 199
- contour smoothing · 199
- contours
 - blank · 403
 - grid node editor · 406
- convergence · 125
- convert · 408–10
- convert grids · 410
- convert Z values · *See* math
- coordinates · 585
 - base maps · 237
 - bitmap · 236
 - exported · 465
- copy · 375, 506
- copy and paste · 461
- copy to worksheet
 - statistics · 71
- correlogram · 438
- cosine · 562
- count · 145
- count property · 500
- cover map · 238
- create several maps in the same plot window · 374
- CreateObject · 494, 502
- creating a color spectrum file · 369
- creating a variogram · 171
- creating anchor nodes · 365
- creating dialog boxes · 528
- creating grid files · 91
- creating maps · 6
- critical value of the K-S statistic · 80
- crop · *See* limits or blanking
- cross axis at · 351
- cross section · 443, 455, 454–57
 - accumulated horizontal distance · 454
 - blanking file · 455
 - boundary number · 454
 - data file contents · 454
 - plotting · 456
- cross validate · 100–103
 - references · 103
 - statistics · 102
 - steps · 100
- cross validation results file · 102
- CSV · 65, 82, 593
- CTRL+A · 476
- CTRL+click · 475, 477
- cubic convolution resampling · 399, 401
- cubic surface · 131
- cubic variogram model · 180
- currency · 66, 515
- current date · 85
- current session only · 553
- current setting file · 552, 559
- current time · 85
- curvature · 425
 - plan · 431
 - profile · 430
 - tangential · 432
- curve tolerance · 223
- custom colors · 481
- custom dialog box · *See* dialog box
- custom fill pattern · 481
- custom limits · 331
- custom line style · 481
- custom setting files · 559
 - creating · 559
- cut · 63, 445, 446, 485, 506
- cut and fill volume · 443, 446
- cut minus fill · 445
- cycle spacing · 340
- cyclic wrap edge of grid · 397

D

- DAT · 22, 65, 82, 239, 409, 454, 593
- data · 18–22, 23, 53–56, 561
 - 2-grid vector map · 275–78
 - convert numeric to text · 56
 - duplicate · 92–95
 - filter · 92–95
 - honoring in grid file · 150
 - numeric · 56
 - opening in worksheet · 61

Index

- post · 42
- regularly spaced · 158
- sort · 67
- statistics · 70
- text · 56
- transform · 69, 561, 565
- data as numbers or text · 55
- data columns · 19
- data exclusion filter · 94
 - variogram · 173
- data file · 18–22
 - change · 247
 - creating · 21
 - cross section · 454
 - save · 22
- data file formats · 61, 593–94
- data filename · 247, 249
- data files · 6, 53, 243
 - ASCII · 82
 - symbol specifications · 244
- data labels · 244
- data limits · 88
- data location statistics · 144
- data metrics · 139–46, 151
 - data location statistics · 144
 - data set · 139
 - description of options · 142–46
 - horizontal planar grid · 140
 - other Z statistics · 144
 - planar grid · 140
 - setting options · 141
 - Z moment statistics · 143
 - Z order statistics · 142
- data page
 - new variogram · 172
 - vector map · 280, 281
- data points
 - posting · 241
- data repeatability · 164, 186
- data statistics
 - terrain statistics · 146
- data to color mapping · 366, 367, 369, 370
- data type list · 533

- data value for symbol size
 - post map · 257
- date · 468, 515, 572
- DBF · 592
- DCX · 232, 585, 589
- DDF · 89, 160, 197, 232, 588
- debug
 - add watch · 538
 - browse · 495, 533
 - clear all breaks · 539
 - quick watch · 509, 538
 - set next statement · 539
 - show next statement · 539
 - step into · 538
 - step out · 539
 - step over · 539
 - step to cursor · 539
 - toggle break · 539
- Debug.Print · 509, 537
- debugging scripts · 536–40
- decimal digits · 35, 46, 66, 359, 480, 558
- default angle · 247
- default gridding method · 91, 150
- default linear variogram
 - defined · 193
 - nugget effect · 193
 - slope · 193
- default method
 - variogram autofit · 189
- default path · 548
- default R^2 · 134
- default scale · 337
- default settings
 - formats for attribute values · 554
- default symbol · 248
- default text properties · 468
- defaults · 547–59
- Delaunay triangulation · 127, 135
- delete · 63, 485
 - delete anchor nodes · 366
 - delete contour label · 225
 - delete label · 32
 - delete levels · 203

- delete method · 501
- delete objects · 492
- delete overlay · 380
- delimiters · 82
- DEM · 89, 160, 197, 259, 268
- density
 - grid · 95, 97
- density of data · 241
- derivatives · 421–26
- derived objects · 499
- descending sort · 68
- deselect all · 478
- deselect cells · 62
- detrend · 174
- device independent bitmap · 462
- dialog box
 - creating · 528
 - example · 531
 - show in script · 529
- dialog defaults · 552–59, 595
- dialog function · 529
- dialog title · 46
- difference between Z value and grid · 457
- differential and integral operators · 433–36
 - biharmonic operator · 435
 - gradient operator · 434
 - integrated volume · 435
 - Laplacian operator · 434
- diffuse · 322
- digital elevation models · 160
- digital terrain elevation data · 161
- digitize · 86, 87, 239, 240, 455
- dim · 515
- direction · 274
 - 2-grid vector map · 275
 - variogram · 168
- directional curvature · 425
- directional derivatives · 421–26
 - curvature · 425
 - definition · 422
 - first derivative · 423
 - graphical example · 426
 - second derivative · 424
- display a message box · 548
- display properties · 478
- distance
 - contour labels · 223
- distance to farthest · 145
- distance to nearest · 145
- distance weighting filter · 391
- distribution of data · 241
- dither bitmap · 263
 - shaded relief map · 272
- division · 561
- DLG · 232, 585
- do not detrend data · 174
- do...loop · 518
- dock
 - object manager · 490
- document object · 499, 503, 504, 505
- documents collection · 503, 542
- Documents collection · 500
- documents property · 503
- don't draw (blank) · 318
- dots per inch · 463, 464
- double · 515
- double quotes · 82
- down and then across · 85
- draw · 467–74
 - ellipse · 474
 - polygon · 469
 - polyline · 470
 - rectangle · 472
 - rounded rect · 473
 - symbol · 471
 - text · 468
- drawing page
 - preferences · 549
- drift · 118, 119
- DT* · 89, 197
- DT1 · 161
- DT2 · 161
- DTED · 89, 161, 197
- duplicate data · 92–95
 - variogram · 172
 - X tolerance · 94

Index

Y tolerance · 94
DXF · 37, 230, 232, 466, 585, 592

E

E00 · 232, 585, 592
edge effects · 390
edge of grid · 396–97
edit
 clear · 63
 copy · 375
 cut · 63, 485
 delete · 63, 485
 deselect all · 478
 insert · 64
 invert selection · 476
 object ID · 45, 485
 paste · 60, 375
 paste special · 60, 63, 462
 properties · 171, 200, 233, 247, 261, 269,
 279, 295, 317, 332, 478, 526
 references · 532
 reshape · 481
 select all · 44, 48, 476
 undo · 548
 UserDialog editor · 528
edit axis properties · 33–36
edit class module properties · 526
edit contour labels · 32, 224
edit data · 53–88
edit dialog box · 529
edit labels · 224
edit overlay · 380
edit polygon · 481
edit polyline · 481
edit UserDialog properties · 532
editing
 axes · 347
elevation · 297, 311
eliminate negative Z values · *See* math
eliminating duplicates · 94
ellipse · 474
 search · 106

e-mail · 16
embossing filters · 393
EMF · 232, 589
empty cells · 54
end · 535
end sub · 509
enhanced metafile · 232, 462
enter a function · 417
enter data · 58
equal intervals · 252
equal number · 252
equations · 561, 573
error codes
 worksheet · 566
error variance · 186
errors · 536–40
ESRI ArcInfo export format · 232, 592
ESRI shapefiles · 232, 592
estimator type
 variograms · 176
estimators
 autocorrelation · 178
 autocovariance · 177
 classical variogram · 177
 standardized variogram · 177
evenly spaced data · 158
exact interpolators · 156
examples
 anisotropy · 110
 ASCII grid file · 577
 blanking file format · 587
 class module · 527
 color spectrum file · 584
 custom dialog box · 531
 grid density · 99
 grid line geometry · 96
 gridding · 90
 index contours · 211
 map background · 346
 mathematical functions · 567
 net volume · 446
 properties · 479
 Scripter · 540–46

- subscript · 573
- superscript · 573
- transform · 70, 565
- variogram grid · 167
- vector · 276
- worksheet header and footer · 86
- exclude data within this distance of the
 - validation point · 102
- excluding data · 94
- expand a grid · 385
- expand blanked region
 - filter · 398
- expand grid · 385
- expand new object manager items · 548
- expand object manager list · 491
- experimental curve · 170
- experimental page · 175–80
- exponential · 359
 - label format · 480
- exponential functions · 562
- exponential numeric format · 66
- exponential variogram model · 180
- export · 463, 585
 - bitmap · 463
 - coordinates · 465
 - format suggestions · 466
 - selected objects only · 463
- export contour lines · 37
- export contours · 230
- export files · 461
- export options · 463
- export rectangle · 465
- export variogram · 194
- extended Simpson's 3/8 rule · 447, 448
- extended Simpson's rule · 447
- extended trapezoidal rule · 447
- extension · 81
- extent
 - grid · 95
- extract · 327, 411–12

F

- F1 · 511
- face name · 556
- fault line properties · 202
- faults · 113, 111–13, 202
 - blanking file · 111
 - grid file · 111
 - memory · 113
- fax · 16
- features · 2–4
- feedback · 15
- field of view · 200, 242, 260, 268, 274, 295, 316, 331, 332
- file
 - export · 463, 585
 - import · 61, 461, 585
 - new · 4, 21, 375
 - new module · 525
 - open · 4, 20, 61, 375, 548
 - page setup · 83–86
 - preferences · 375, 476, 559, 547–59
 - recent files · 548
 - save · 6, 22, 36, 81, 463, 548
 - save as · 22, 81, 463
- file extension · 81
- file formats · 81, 575–94
 - ASCII data · 593
 - ASCII data files · 82
 - ASCII delimiters · 593
 - Atlas boundary · 590–91
 - base maps · 232
 - binary grid · 578
 - blanking files · 587
 - color spectrum files · 583
 - data · 593–94
 - GS ASCII grids · 575–81
 - GS binary grid · 578
 - level files · 582
 - Surfer grids · 579–81
- file name · 81, 85
- file type
 - specifying during export · 463

Index

- file types · 6
- fill · 363, 445, 446
 - contour map · 212–17
 - contours · 212
 - custom · 481
- fill blanked region
 - filter · 398
- fill contours · 30, 213, 214
- fill dialog · 213
- fill edge of grid · 397
- fill pattern · 547, 555, 596
 - custom · 481
- fill pattern names · 595
- fill properties · 479
 - base map · 233
 - blanked regions · 202
 - default · 550
 - default settings · 555
- fill properties dialog · 214
- fill spectrum
 - contour map · 213
- filled contour map · 212–17
- filled contours · 201
- filter
 - blank edge · 396
 - blanked nodes · 398
 - coefficient of variation · 394
 - compass gradient · 395
 - cyclic wrap edge · 397
 - distance weighting · 391
 - edge effects · 390
 - edge of grid · 396–97
 - embossing · 393
 - expand blanked region · 398
 - fill blanked region · 398
 - fill edge · 397
 - Gaussian low-pass · 392
 - general user-defined · 392
 - gradient directional · 393
 - grid · 388–98
 - high-pass · 393
 - ignore blanked region · 398
 - ignore edge · 396
 - inverse distance · 392
 - Kirsch · 395
 - leave blanked region · 398
 - linear convolution · 390–93
 - lower quartile · 393
 - low-pass · 391–92, 393
 - median · 393
 - median difference · 395
 - minimum · 393
 - mirror edge · 397
 - moving average · 391
 - neighborhood · 388
 - nonlinear · 393–95
 - number of passes · 390
 - order 1 derivative · 393
 - order 2 derivative · 393
 - predefined · 393
 - Prewitt · 395
 - range · 393
 - references · 398
 - replicate edge · 396
 - Robinson 3-level · 395
 - Robinson 5-level · 395
 - shift and difference · 393
 - standard deviation · 394
 - threshold averaging · 395
 - upper quartile · 393
 - user-defined · 391
 - variance · 394
- filter data · 92–95
 - Boolean filter · 94
- filter size · 390
- filters · 389
- first · 93, 209, 214, 221, 227
- first derivative · 423
- first input row · 72
- first major tick · 355
- first quartile · 73
- fit criteria
 - variogram autofit · 189
- fit to · 83
- fit to window · 25
- fixed · 359

- label format · 480
- fixed numeric format · 66
- fixed size · 43, 248
- fixed symbol size
 - post map · 256
- fixed symbols · 241
- flat lighting · 321
- flow control · 518
- follow value · 533
- font · 235, 468
 - axis tick labels · 357
 - labels · 251
 - size · 556
- font properties
 - default · 550
 - default settings · 556
- font size · 570
- font, axis title · 352
- footer
 - worksheet · 85
- footers · 84
- for each...next · 519
- for...next · 519
- forecolor · 555
- foreground color · 31, 555
- format
 - cell properties · 66
 - column width · 64
 - data files · 53
 - labels · 251
 - row height · 65
- formats for attribute values · 554
- formula calculation · 561
- formulas · 561
- forum · 15
- Fourier and spectral analysis · 437–41
 - correlogram · 438
 - periodogram · 439
- Fourier frequency · 441
- frame
 - legend · 254
 - vector map legend · 291
- free rotate · 483, 488

- frequency
 - classed post map symbols · 249
 - contour labels · 221
 - contour line properties · 209
 - filled contours · 214
 - hachures · 227
 - post map symbols · 248
 - vector map symbols · 283
- from edge · 85
- function · 159, 561
- function statement · 522
- function type · 134
- function, grid
 - Boolean expressions · 94
- functions · 69, 520, 522, 539
- functions button · 69
- functions of an interval of columns · 563

G

- Gaussian low-pass filter · 392
- Gaussian variogram model · 180
- general numeric format · 66
- general page
 - axis properties · 357
 - classed post map · 249, 254
 - contour · 201
 - image map · 261
 - new variogram · 173
 - post map · 247
 - preferences · 547
 - shaded relief map · 269
 - surface · 317
 - wireframe · 296
- general recommendations
 - gridding · 150
- general user-defined filter · 392
- get variogram · 117, 194
- GetFilePath function · 524
- GetObject · 502
- GIF · 232, 466, 585, 589
- global variables · 517
- glyph number · 245

Index

- Golden Software blanking file · 232, 466
- Golden Software blanking file format · 586–87
- Golden Software boundary file · 232, 586
 - spatial file format · 466
- Golden Software data · 22, 82
- Golden Software interchange · 232, 588
- Golden Software PlotCall · 232, 587
- gradational colors · 364
 - contour map · 212–17
- gradational line color · 364
- gradational line properties · 207
- gradient directional filters · 393
- gradient method
 - shaded relief map · 271
- gradient operator · 434
- Grapher · 456
- graphics
 - importing · 461
- GRD · 24, 160, 197, 409, 575–81
- Greek characters · 468
- Greek letters · 569
- grid · 361
 - ASCII · 409
 - binary · 409
 - blank · 216, 404, 403–5
 - Boolean expressions · 94
 - calculus · 418–41
 - convert · 408–10
 - creating a denser grid · 385
 - data · 23, 91, 94, 114, 561
 - defined · 89
 - drawing · 551
 - extract · 327, 411–12
 - filter · 388–98
 - function · 94, 159, 561
 - grid node editor · 404, 405, 406–8
 - math · 94, 277, 416–18, 561
 - mosaic · 399–402
 - number of lines · 95
 - object · 494
 - residuals · 457–59
 - slice · 454, 455
 - spacing · 95
 - spline smooth · 383–87
 - standard deviation · 119
 - transform · 412–15
 - variogram · 94, 171, 194, 561
 - volume · 436, 443–53
- grid calculus · 418–41
 - references · 442
- grid columns · 95
- grid density · 95, 400
 - defining · 97, 98
 - extract · 98
 - shaded relief map · 267
 - spline smooth · 98
- grid divisions · 551
- grid extents · 400
- grid file · 22–24, 261, 269, 296, 389
 - angle (direction) component · 277
 - change · 201, 280, 281, 318
 - create · 22–24
 - density · 446
 - length (magnitude) · 277
 - volume · 446
 - wireframe · 294
 - X component · 275
 - Y component · 275
- grid file formats · 575–81
 - binary · 578
 - GS ASCII · 575–81
 - GS binary · 578
 - Surfer 7 grid · 579–81
 - Surfer grid · 579–81
- grid files · 6, 97, 156, 160, 503
 - creating · 91
 - faults · 111
 - formats · 89
 - volume between · 443
- grid filter · 388–98
- grid function · 159
- grid info · 201, 261, 269, 296, 318, 389, 399, 412
 - grid node editor · 408
- grid limits · 95, 293, 400

- defining · 97
- grid line geometry · 95–99, 400
 - example · 96
- grid lines · 360
 - default number · 95
- grid menu · 7
- grid node · 89
 - weighted average · 157
- grid node editor · 405, 406–8
 - blank node · 407
 - contour levels · 407
 - show contours · 406
 - show nodes · 406, 407
 - unblank node · 408
- grid node editor window · 4
- grid nodes · 198, 406
 - blanking · 403–5
- grid operations · 383
- grid report · 103
- grid residuals · 458, 459
- grid rows · 95
- grid spacing · 98
- grid utilities · 7
- grid volume report · 443, 445
- gridding · 6, 7
 - breaklines · 112
 - canceling · 98
 - cross validate · 100–103
 - example · 90
 - faults · 113
 - filter data · 92–95
 - functions · 159
 - large data sets · 155
 - minimum number of points · 92
 - moderate-sized data sets · 155
 - overview · 89
 - regularly spaced data · 158
 - small data sets · 155
 - view data · 95
- gridding methods · 89, 114–49
 - choosing · 150–57
 - comparing · 150
 - comparison · 151

- data metrics · 139–46
- default · 150
- exact interpolators · 156
- inverse distance to a power · 114–16
- kriging · 117–21
- local polynomial · 147–49
- minimum curvature · 121–25
- modified Shepard's method · 125–27
- moving average · 137–38
- natural neighbor · 127–29
- nearest neighbor · 129–30
- polynomial regression · 131–32
- radial basis function · 132–35
- recommendations · 150
- smoothing interpolators · 156
- triangulation with linear interpolation · 135–37

- gridding options · 103
- gridlines
 - worksheet · 85
- GridWindow · 500
- GS ASCII · 160, 409
- GS ASCII grid file · 575–81
- GS binary · 160, 409
 - grid file format · 578
- GS Surfer_2 · 462
- GSB · 232, 466, 585, 586
- GSI · 232, 585, 588
- GTopo30 · 89, 160

H

- hachure closed contours only · 226
- hachures · 199, 225–28
 - removing · 228
 - specific levels · 227
- handle size · 549
- handles · 477
 - hollow · 477
 - solid · 477
- HDR · 89, 160, 197
- header
 - worksheet · 85

Index

- header row · 20
- headers · 84
- headers and footers
 - examples · 86
- help · 15
 - about Didger · 16
 - automation help · 495, 499
 - basic language help · 511
 - check for update · 14
 - contents · 511
 - feedback · 15
 - help topics · 15
 - Surfer automation help · 495, 499, 511
- help button · 586
- help string · 534
- help topics · 15
- hidden lines · 378
- hide areas of a map · 238
- hide map · 380
- hierarchy · 496
- highlight objects · 475–77
- high-pass filters · 393
- histogram
 - variogram · 191
- hollow handles · 477
- honoring data · 150, 156
- horizontal distance
 - cross section · 454
- horizontal light position
 - shaded relief map · 270
 - surface · 322
- horizontal planar grid · 140
- horizontally align maps · 381

I

- ID · 45
- ID object · 485
- identity · 483, 489
- if function · 94, 564
- if...else...end if · 518
- if...end if · 518
- ignore blanked region

- filter · 398
- ignore case · 69
- ignore data outside grid · 95
- ignore edge of grid · 396
- image coordinates · 234, 236
- image map · 9, 259–64, 363
 - add color scale bar · 264
 - color scale bar · 263, 264–65
 - color scale bar visibility · 264
 - color spectrum · 363
 - colors · 263, 364
 - components · 260
 - creating · 260
 - dither bitmap · 263
 - edit color scale bar · 265
 - general page · 261
 - input grid · 261
 - interpolate pixels · 262
 - missing data · 263
 - overlays · 264
 - properties · 259–64
 - remove color scale bar · 264
 - show color scale · 264
 - USGS DEM · 259
- immediate window · 509
- import · 61, 461, 585
 - maps and graphics · 461
- import options · 462
- inches · 550
- index
 - symbol · 244, 557
- index contours · 211
- initial
 - variogram autofit · 190
- input grid · 261
- input grid file · 201, 234, 269, 280, 317, 389, 412
- input grid file A · 416
- input grid file B · 417
- input grid files · 399
- InputDialog function · 524
- insert · 64
- insert grid nodes · 385

insert levels · 206
insert nodes · 385
installation
 network · 13
 troubleshooting · 14
installation directions · 12
integer · 515
integrated volume · 435
internal default · 553
internal tension · 123, 124
interpolate pixels · 262
interpolation · 156
 weighted average · 157
interquartile range · 142
interruptible redraw · 550
interval · 27, 205
inverse distance filter · 392
inverse distance to a power · 114–16, 150,
 156, 157
 equations · 115
 references · 116
 setting options · 115
 smoothing · 115
 weighting power · 115
inverse multiquadric · 134
invert selection · 476
invisible contour lines · 217
isopach map · 417
italic · 570
italic text · 12
item method · 500, 504

J

join grids · *See* mosaic
JPEG · 464
JPEG compressed bitmap · 232, 589
JPG · 232, 464, 466, 585, 589

K

kernel functions · 134

key words · 509
keyboard
 simulating mouse operations · 487
keyboard commands
 worksheet · 58–60
Kirsch filter · 395
Kolmogorov-Smirnov goodness of fit · 79
kriging · 117–21, 150, 155, 156, 157
 block · 119
 get variogram · 194
 point · 119
 references · 120
 setting options · 117
 standard deviations · 119
 using variogram results in · 194
 variogram components · 164
 variograms · 118

L

label
 add · 32
 delete · 32
 font · 251
 format · 251
 move · 32
label angle · 250
label format · 46, 359, 480
 absolute value · 480
 axis tick labels · 357
 compact · 480
 exponential · 480
 fixed · 480
 prefix and suffix · 480
 thousands separator · 480
label increment · 340
label plane · 250
label properties · 480
 axis tick labels · 357
 default settings · 558
 scale bar · 340
label to edge distance · 224
label to label distance · 223

Index

- labels · 32. *See* contour labels
 - data · 244
 - digits · 35
 - in worksheet · 55
 - math text · 569
 - post map · 46, 242
 - post maps · 250–51
- labels in first row · 69, 71
- labels page · 250
- lag direction · 175
- lag width · 178
 - smoothing variograms · 178
- lake · 452
- Lambertian reflection · 271
- landscape · 83
- Laplacian operator · 434
- last · 93
- last input row · 72
- last major tick · 355
- latitude · 338
- least absolute value
 - variogram autofit · 189
- least squares
 - variogram autofit · 189
- leave blanked region
 - filter · 398
- left alignment
 - worksheet · 67
- left axis · 347
- left justified · 56
- left/center/right separator · 85
- legend · *See also* color scale bar
 - classed post map · 241, 249, 254–56
 - edit properties · 254
 - frame · 254
 - margins · 254
 - object · 499
 - reverse order · 256
 - symbol size · 256
 - template · 255
 - title · 254
 - vector map · 274, 285, 290–91
- legend properties dialog · 254
- length
 - scale · 337
 - variogram · 170, 185
 - variogram model · 184
- length (A)
 - variogram · 180
- length (magnitude) data
 - vector map · 277
- length component grid file · 281
- length grid · 278
- length scale · 164
- level 0 files · 161
- level 1 files · 161
- level 2 files · 161
- level file · 299, 301, 363, 370
 - creating · 371
 - property names · 595
- level file format · 582
- level properties · 370
- levels · 25, 204, 299, 300
 - add · 203, 206
 - change properties · 28
 - color fill · 30
 - delete · 203
 - wireframe · 304
- Levels collection · 500
- levels page · 203, 371
- LGO · 232
- LGS · 232
- library list · 533
- light color · 322
- light position angles
 - shaded relief map · 270
 - surface · 321
- light source · 268, 316
 - shaded relief map · 270
 - surface · 321
- light source direction
 - shaded relief map · 267
- lighting · 321
- lighting page · 321
- limits · 198, 200, 242, 260, 268, 274, 316, 331, 342–43

- axis · 350
 - axis scaling · 343
 - base map · 231, 237, 343
 - clip vectors at · 288
 - grid · 95, 97
 - map · 337
 - overlays · 343, 377
 - post maps · 343
 - scale · 343
 - use data limits · 344
 - vector map · 288
 - wireframe · 293, 342
 - line
 - custom · 481
 - variogram model · 192
 - line continuation · 514
 - line properties · 479
 - axis · 360
 - base map · 233
 - blanked regions · 202
 - contour map · 29, 199, 207–11
 - default · 550
 - default settings · 555
 - gradational · 207, 364
 - scale bar · 340
 - specific contour levels · 210
 - uniform · 209
 - wireframe · 40, 299, 300, 302, 303–6, 308
 - line property precedence · 303
 - line props · 297, 311
 - line spacings · 95
 - line spectrum · 40
 - line style names · 595
 - linear convolution filters · 390–93
 - linear detrending · 174
 - linear drift · 118, 119
 - linear model · 185
 - slope · 185
 - linear resampling
 - surface · 324
 - linear scaling
 - vector map · 284
 - linear symbol scaling · 257
 - linear variogram · 118
 - defined · 193
 - nugget effect · 193
 - slope · 193
 - standard deviation grid · 119
 - linear variogram model · 180
 - lines · 198
 - wireframe · 301
 - load a color spectrum · 366
 - loaded window · 509, 540
 - local polynomial · 147–49, 151
 - equations · 147–48
 - setting options · 149
 - log file · 14
 - logarithm · 563
 - logarithmic scaling
 - vector map · 284
 - logarithmic variogram model · 180
 - logical operators · 94, 564
 - Lommel-Seeliger law · 271
 - long · 515
 - longitude · 338
 - Lotus · 594
 - lower color
 - surface · 318
 - lower quartile · 142
 - lower quartile filter · 393
 - lower surface · 297, 443, 445, 446
 - low-pass filter · 391–92
 - low-pass filters · 393
 - LVL · 203, 299, 301, 370–71, 582
-
- M**
- M.A.D. · 144
 - MacIntosh bitmap · 232
 - magnitude · 274
 - 2-grid vector map · 275
 - magnitude (data)
 - vector map · 285
 - mailing address · 16
 - major interval · 35, 350, 355
 - major ticks · 354

Index

- editing · 355
- spacing · 355
- starting and ending · 349
- map
 - base map · 231–36, 237, 238, 461, 585
 - break apart overlay · 380
 - change order overlay · 482
 - contour map · 25, 32, 197, 200, 224, 230
 - digitize · 86, 87, 239, 240, 455
 - image map · 259–64
 - orientation · 334, 335
 - overlay maps · 44, 48, 237, 332, 373, 376–80
 - post map · 42, 55, 245, 246, 456
 - scale · 339
 - scale bar · 340
 - shaded relief map · 267–72, 267–72
 - stack maps · 373, 375, 381
 - surface · 47, 316
 - trackball · 38, 332, 335
 - vector map · 273–91
 - wireframe · 37, 295
- map axes · 200, 242, 260, 268, 274, 295, 316, 347
- map background example · 346
- map coordinates · 239, 465
- map ID · 485
- map limits · *See* limits
- map menu · 6
- map objects · 504
- map orientation · 200, 242, 260, 268, 274, 295, 316
- map properties · 25
 - opening · 332
- map scale · *See* scale
- map series
 - color · 367
- map types · 7
- MapFrame object · 507, 508, 545, 546
- MapInfo interchange format · 592
- MapInfo interchange format file · 232
- maps
 - arranging · 373
 - combining · 373
 - creating · 6
 - moving · 374
 - multiple · 374
 - positioning · 373
- margins
 - legend · 254
 - show · 550
 - worksheet · 84
- markers · 471
- mask · 238
- mask nodes · *See* blanking
- material color · 50, 318
- math · 277, 416–18, 561
 - Boolean expressions · 94
 - isopach map · 417
- math in the worksheet · 564
- math text · 569–73
 - bold · 570
 - color · 571
 - date · 572
 - examples · 573
 - font size · 570
 - italic · 570
 - strikethrough · 570
 - subscripts · 571
 - superscripts · 571
 - symbol · 572
 - time · 572
 - typeface · 570
 - underline · 570
- math text instruction syntax · 569
- math text instructions · 468
- mathematical equations · 561
- mathematical functions · 561
 - examples · 567
- mathematical symbols · 569
- mathematical transformations
 - worksheet · 69
- max lag dist · 178, 179
- max lag distance · 173
 - variogram autofit · 190
- max total order · 131

- max X order · 131
- max Y order · 131
- maximum · 142
 - anchor node value · 366
 - variogram autofit · 190
- maximum axis · 350
- maximum distance
 - variogram autofit · 190
- maximum filter
 - maximum · 393
- maximum grid values
 - in color spectrums · 368
- maximum iteration · 123, 125
- maximum iterations
 - variogram autofit · 189
- maximum number of data to use from all sectors · 105
- maximum number of data to use from each sector · 105
- maximum residual · 123, 125
- maximum rows and columns · 88
- maximum value · 72
- maximum X · 93
- maximum Y · 93
- maximum Z · 93
- mean · 72, 143
- median · 72, 142
- median absolute deviation · 144
- median difference filter · 395
- median distance · 145
- median filter · 393
- median X · 93
- median Y · 93
- median Z · 93
- memory
 - faults · 113
 - surface · 327
- memory requirements
 - worksheet · 88
- merge grids · 399–402
- mesh · 319
- mesh lines · 49, 316
- metafiles · 589
- methods
 - automation · 495
 - defined · 523
- methods/properties list · 533
- metrics · 139–46
- micro variance · 186
- Microsoft Excel · 594
- Microsoft SYLK · 594
- MID · 592
- middle value · 72
- midpoint difference · 271
- midrange · 93, 142
- MIF · 232, 585, 592
- minimum · 142
 - anchor node value · 366
 - variogram autofit · 190
- minimum axis · 350
- minimum curvature · 121–25, 150, 155
 - boundary tension · 124
 - convergence · 124
 - equations · 122
 - internal tension · 124
 - maximum iteration · 125
 - maximum residual · 125
 - references · 125
 - relaxation factor · 124
 - setting options · 123
- minimum filter · 393
- minimum grid values
 - in color spectrums · 368
- minimum number of data · 138
- minimum number of data in all sectors · 105
- minimum value · 72
- minimum X · 93
- minimum Y · 93
- minimum Z · 93
- minor ticks · 356
 - displaying · 356
- minor ticks per major · 356
- minus · 561
- mirror edge of grid · 397
- mirror X transform · 415
- mirror Y transform · 415

Index

miscellaneous functions · 563
missing data · 54, 263
 shaded relief map · 272
mode · 555
model curve · 170
model line
 variogram model · 192
model page · 180–87
 autofit · 188–90
modified Shepard's method · 125–27, 150,
 156, 157
 quadratic neighbors · 127
 references · 127
 setting options · 126
 weighting neighbors · 127
modify axis · 33–36
modify individual levels · 206
modifying grid files · 383
module properties · 526
module types · 525
modules · 524
monochrome · 464
mosaic · 399–402
 add grid · 399
 bilinear interpolation · 401
 blanked node · 402
 calculate from input extents · 400
 cubic convolution · 401
 grid extents · 400
 grid line geometry · 400
 nearest neighbor · 401
 overlap method · 399
 remove grid · 399
 resample methods · 399, 401
 seamless grids · 402
 tips · 402
move backward · 482
move contour labels · 224
move forward · 482
move label · 32
move to back · 482
move to front · 377, 482
moving average · 137–38, 151

 setting options · 137
moving average filter · 391
moving maps · 374
moving objects · 486
moving the active cell · 58
multilog · 134
multi-page PCX · 589
multiple maps per page · 374
multiplication · 561
multiquadric · 132, 134
multithreaded redraw · 550

N

named arguments · 520
names
 colors · 595
 fill patterns · 595
 line style · 595
naming objects · 485
national imaging and mapping agency · 161
natural cubic spline · 134
natural logarithm · 563
natural neighbor · 127–29, 150, 156
 references · 129
 setting options · 128
nearest neighbor · 129–30, 151, 156, 158
 setting options · 130
nearest neighbor resampling · 399, 401
nearest resampling
 surface · 324
negative planar area · 445, 448
negative surface area · 445, 449
negative values
 removing from grid · 418
 vector map polar grids · 277
negative volume · 445, 446
neighborhood · 388
nested variogram model · 180
net volume · 443, 446
 example · 446
network installation · 13
new · 21, 375

- new classed post map · 246
- new contour map · 25, 197, 200
- new features · 2–4
- new module · 511, 525
- new post map · 42, 245, 456
- new script · 511
- new variogram · 172–74
 - data page · 172
 - general page · 173
- NewWindow · 504
- NIMA · 161
- no data areas · *See* blanking
- no search (use all of the data) · 104, 105
- node · 89
- node is blanked if fewer · 105
- nominal image size · 463
- none · 93
- non-linear filters · 393–95
- non-numeric entries · 54
- not · 94, 564
- nugget effect · 168, 169, 180, 184, 185
 - default linear variogram · 193
 - error variance · 186
 - micro variance · 186
 - smoothing · 185
- null value · *See* blanking value
- number format · 480
- number formatting · 66
- number nodes to insert · 385
- number of classes · 252
- number of columns and rows · 95
- number of cycles · 340
- number of decimal digits · 558
- number of lags · 178, 179
- number of lines · 95
- number of missing values · 72
- number of passes · 390
- number of random points to validate · 102
- number of sectors to search · 105
- number of values · 72
- numbers · 55
- numeric data limits · 88
- numeric format · 558

- numeric label properties
 - default settings · 558

O

- object · 515
 - accessing · 494
 - application · 494
 - attributes (properties) · 494
 - axis · 496
 - base · 500
 - collection · 500
 - document · 499
 - grid · 494
 - legend · 499
 - properties · 478–81
 - services (methods) · 494
 - shape · 499
 - shapes · 496
 - store · 523
 - window · 499
 - wireframe · 496
- object browser · 533
- object hierarchy · 496, 499, 523
- object ID · 45, 485, 491
- object list · 509
- object manager · 4, 42, 45, 377, 475, 485, 490–92, 548
 - arranging objects · 492
 - docking · 490
 - expand and collapse list · 491
 - float · 490
 - object ID · 491
 - opening and closing · 490
 - select objects · 491
 - visible status · 490
- object model · 496
 - Surfer 6 · 508
- object module · 511, 525, 526
- object variables · 516
- objects · 527
- objects · 499, 494–508, 533
 - collection · 495

Index

- derived · 499
- using · 523
- observation
 - variograms · 176
- offset
 - axis tick labels · 357
- offset along axis · 352
- offset distance · 145
- offset from axis · 352
- offset transform · 413
- one grid vector map · *See* vector map
- opaque · 213
- open · 4, 61, 375, 548
- open data file · 20
- open map properties · 332
- open method · 503
- open properties · 478
- open Surfer · 18
- open uses · 511
- opening data files · 61
- operation
 - transform · 413
- operators · 518, 564
 - in formulas · 561
- optional arguments · 519
- options
 - blank node · 407
 - contour levels · 407
 - grid info · 408
 - grid node editor · 406, 408
 - gridding · 103
 - show contours · 406
 - show nodes · 406, 407
 - unblank node · 408
- or · 94, 564
- order 1 derivative filters · 393
- order 2 derivative filters · 393
- order objects · 477
- order of transformations · 489
- order overlay · 377, 482
- orientation · 200, 242, 260, 268, 274, 295, 316, 334, 335
- orthographic · 333
- other Z statistics · 144
- output BLN file · 455
- output DAT file · 455
- output grid file C · 417
- output grid of kriging standard deviations · 117
- overlap method · 399
- overlapping maps · 374
- overlay adjacent maps · 378
- overlay maps · 44, 237, 332, 373, 376–80
 - change order · 482
- overlay resolution · 324
- OverlayMaps method · 545
- overlays · 506
 - adjacent · 378
 - arranging · 377
 - delete · 380
 - edit · 380
 - hide · 380
 - image map · 264
 - limits · 377
 - limits and scale · 343
 - raster · 373
 - select map · 45
 - shaded relief map · 272
 - surface · 47, 315, 316, 323, 327
 - visibility · 324, 380
 - wireframe · 373
 - wireframes · 313
- overlays collection · 495, 507
- Overlays collection · 500
- overlays property · 507
- overview
 - how to use Surfer · 18

P

- page coordinates · 465
- page number · 85
- page order · 85
- page outline · 550
- page rectangle · 465
- page setup

- worksheet · 83–86
- page units · 550
- pair comparison files · 165
- pairs · 170
 - variogram model · 192
- parameter fit limits · 190
- parameters list · 534
- parent property · 501
- parentheses
 - Scripter · 521
- paste · 60, 375, 461, 506, 533
- paste special · 60, 63, 462
- pattern · 555
- patterns
 - fill properties · 479
 - printing · 547
- pause · 538
- PCT · 232, 466, 585, 589
- PCX · 232, 585, 589
- PCX (ZSoft/Paintbrush) · 232, 589
- pentaspherical variogram model · 180
- periodogram · 439
- perspective · 333
- Peucker's approximation · 271
- PICT (Macintosh) bitmap · 589
- PICT MacIntosh bitmap · 232
- picture · 462
- pixels · 463
- plain · 570
- plan curvature · 431
- planar area · 443, 445, 448–50
- planar grid · 140
- plane
 - label · 250
- play a sound · 548
- plot lines of constant · 296, 301, 302
- plot page
 - variogram · 192
- plot window · 4, 6
- PlotCall · 587
- PlotDocument object · 504
- PlotWindow · 500
- PLT · 232, 585, 587
- PNG · 232, 589
- point kriging · 117, 119
- polar coordinates
 - vector map · 281
- polar data · 273, 277
- polygon · 235, 461, 469
 - base map · 233
 - reshape · 481
- polyline · 235, 470
 - custom · 481
 - reshape · 481
- polynomial · 147–49
- polynomial regression · 131–32, 151, 155, 157
 - references · 132
 - setting options · 131
- population statistics · 71
- portable network graphics bitmap · 232, 589
- PortInt · 515
- portrait · 83
- position · 488
 - axis · 351
- position maps · 373
- position relative to symbol · 250
- positional arguments · 519
- positioning an anchor node · 366
- positive planar area · 445, 448
- positive surface area · 445, 449
- positive volume · 445, 446
- post map · 8, 241, 456
 - 3D labels lines · 251
 - angle · 247
 - change data file · 253
 - components · 242
 - create · 245
 - creating · 42
 - data files · 55, 243
 - data values for symbol size · 257
 - default symbol · 248
 - default symbol angle · 248
 - fixed symbol size · 256
 - general page · 247
 - labels · 46, 55, 250–51

Index

- limits · 343
- linear symbol scaling · 257
- properties · 42–43, 247–53
- proportional symbol size · 256
- proportionally scaled symbols · 244
- scaled symbols · 256
- square root symbol scaling · 257
- symbol · 247
- symbol frequency · 248
- symbol height · 257
- symbol number · 245
- symbol size · 248, 256
- symbols from data file · 244
- updating data · 253
- Powell's search method · 188
- power · 391, 392
- power variogram model · 180
- precedence
 - line properties on wireframes · 303
- precedence in calculations · 561
- predefined filters · 393
- preferences · 375, 559, 547–59
 - aperture · 549
 - backup files · 547
 - centimeters · 550
 - default path · 548
 - dialog defaults · 552
 - drawing · 549
 - fill, line, symbol, and text properties · 550
 - grid · 551
 - grid divisions · 551
 - handle size · 549
 - inches · 550
 - interruptible redraw · 550
 - margins · 550
 - multithreaded redraw · 550
 - page outline · 550
 - recent files · 548
 - rectangle must fully surround · 476
 - ruler divisions · 551
 - rulers · 551
 - show position · 551
 - simulate stock hatch patterns · 547
 - snap to ruler · 551
 - undo levels · 548
- prefix · 359, 558
 - label format · 480
- preserve data values · 150
- Prewitt filter · 395
- print
 - surface · 327
- printing patterns · 548
- printing the worksheet · 83
- prisms · 444
- private keyword · 526
- proc (procedure) list · 509
- procedures · 520, 522, 539
- profile · 454
- profile curvature · 430
- projection · 331, 332, 333
- properties · 25, 171, 200, 247, 295, 332, 478–81, 491, 526, 585
 - automation · 494
 - axis · 33–36
 - base map · 233
 - change in base map · 235
 - classed post map · 247–53
 - composite object · 461
 - contour lines · 207–11
 - default · 550
 - defined · 523
 - dialog box · 478
 - fill · 479
 - image map · 259–64
 - label · 480
 - legend · 254
 - line · 479
 - multiple object tips · 479
 - multiple objects · 479
 - overlays · 49
 - post map · 42–43, 247–53
 - post map labels · 46
 - shaded relief map · 269
 - single object · 478
 - surface · 47–51, 326
 - symbol · 479

- text · 479
- vector map · 279, 285
- wireframe · 37–41, 296–301
- property get · 527
- property let · 527
- property names · 595
- property set · 527
- proportional · 248
- proportional scaling dialog · 256
- proportional symbol size
 - post map · 256
- proportional XY scaling · 338
- proportionally scale symbols · 244
- proportionally sized symbols · 241
- public · 527
- public keyword · 517
- public statement · 526

Q

- quadratic
 - drift · 119
- quadratic detrending · 174
- quadratic drift · 118
- quadratic model · 184
- quadratic neighbors · 126, 127
- quadratic surface · 131
- quadratic variogram model · 180
- quick watch · 509, 538

R

- R.M.S. · 144
- R^2 · 133, 134, 157
- radial basis function · 132–35, 151, 155, 156, 157
 - function types · 134
 - inverse multiquadric · 134
 - multilog · 134
 - multiquadric · 134
 - natural cubic spline · 134
 - references · 135

- setting options · 132
- thin plate spline · 134
- radial divisions · 173
- radius
 - linear variogram model · 185
- radius 1 · 138
 - search ellipse · 106
- radius 2 · 138
 - search ellipse · 106
- random · 93
- range · 72, 142, 168
 - axis · 350
 - grid · 95
 - variogram · 184
- range filter · 393
- range object · 505
- raster image · 259
- raster maps · 259
 - overlying · 373
- raster profile · 160
- ratio
 - anisotropy · 109, 187
 - variogram · 185
- rational quadratic variogram model · 180
- recalc grid · 387
- recalculate grid · 386
- recent files · 548
- recommendations
 - gridding · 150
- rectangle · 472
- rectangle must fully surround · 476, 549
- redraw · 550
 - interruptible · 550
- reference vectors · 291
- references
 - cross validate · 103
 - filter · 398
 - grid calculus · 442
 - inverse distance to a power · 116
 - kriging · 120
 - minimum curvature · 125
 - modified Shepard's method · 127
 - natural neighbor · 129

Index

- polynomial regression · 132
- radial basis function · 135
- Scripter · 535
- shaded relief map · 272
- statistics · 80
- terrain modeling · 433
- transform · 489
- triangulation with linear interpolation · 137
- type library · 532
- variogram · 195
- volume · 449
- reflectance
 - shaded relief map · 267
- regularly spaced contour lines · 205
- regularly spaced data · 158
- relative weighting · 108
- relaxation factor · 123, 124
- relief map · *See* shaded relief map
- relief parameters
 - shaded relief map · 271
- remap to · 318
- remove all contour labels · 224
- remove grid · 399
- remove hidden lines · 298
- rename object · 45
- replicate edge of grid · 396
- report · 11. *See* online help
 - grid volume · 445
 - gridding · 103
 - statistics · 95
 - variogram · 191
- resample methods · 401
 - mosaic · 399
 - surface overlay · 324
- reserved keywords · 94, 564
- reshape · 481
- residuals · 443, 457–59
 - defined · 457
 - store · 458
- resize row or column · 64
- resizing objects · 486
 - information in dialog · 487
 - scale · 487
 - transform · 487
- result · 533
- reverse order · 256
- reverse vector orientation · 284
- RGB color · 554
- RGB values · 596
- right alignment
 - worksheet · 67
- right axis · 347
- right justified · 56
- Robinson 3-level filter · 395
- Robinson 5-level filter · 395
- root mean square · 144
- rotate · 332, 483, 488
- rotate maps · 331
- rotate transform · 414
- rotation · 38, 200, 242, 260, 268, 274, 295, 316, 332, 489
- rounded rectangle · 473
- rounded square · 473
- row height · 65
- row numbers · 57
- rows · 505
- ruler divisions · 551
- rulers · 551
- rulers and grid · 375, 551
- run · 535, 538
- running scripts · 535
- run-time errors · 537

S

- same as plot · 256
- sample statistics · 71
- save · 6, 22, 36, 81, 463, 548
- save as · 22, 81, 463
- save as type · 81
- save color spectrum · 367
- save worksheet formatting · 64, 67
- saved scaling source · 465
- saving a map · 36
- saving defaults · 547, 552

- saving Excel files · 81
- scale · 169, 242, 260, 268, 274, 331, 338, 337–39
 - axis · 35, 349
 - base map · 237
 - default · 337
 - defining · 337
 - latitude/longitude · 338
 - length · 337
 - limits · 343
 - linear variogram model · 185
 - map · 199, 294, 316
 - overlays · 343
 - variogram models · 184
 - wireframe · 39
- scale (C)
 - variogram · 180
- scale bar · 340–41
 - cycle spacing · 340
 - example · 341
 - label increment · 340
 - label properties · 340
 - line properties · 340
 - number of cycles · 340
 - tracking · 340
- scale tracks · 340
- scale transform · 413
- ScaleBar objects · 508
- ScaleBars collection · 500, 508
- ScaleBars property · 508
- scaled symbols
 - post map · 256
- scaling method
 - vector map · 284
- scaling page · 284
- scaling source · 465
- scaling worksheet printouts · 83
- screen · 250
- screen layout · 4
- ScreenUpdating · 503
- script · 493
 - end · 535
 - new · 511
 - open · 511
 - pause · 538
 - run · 535, 538
- script runs incorrectly · 537
- Scripter · 5, 508–12
 - '#uses · 525
 - arguments · 520
 - array variables · 516
 - BASIC language · 513
 - class modules · 525
 - code editor window · 509
 - code modules · 525
 - comments · 514
 - debugging · 536–40
 - examples · 540–46
 - flow control · 518
 - functions · 520
 - getting user input · 524
 - global variables · 517
 - help · 511
 - line continuation · 514
 - named arguments · 520
 - object modules · 525
 - object variables · 516
 - operators · 518
 - optional arguments · 519
 - parentheses · 521
 - positional arguments · 519
 - procedures · 520
 - run · 535
 - running scripts from command prompt · 536
 - statements · 514
 - subroutines · 520
 - suggested reading · 535
 - text strings · 514
 - variables · 515
 - VBA compatibility · 534
 - view variable value · 538
 - windows · 509
 - writing functions · 522
 - writing scripts · 512
 - writing subroutines · 521

Index

- scripting · 493–546
 - objects · 494–508
- scripts · 511
- SDTS · 89, 160, 588
- SDTS topological vector profile · 232
- search · 103–8
 - rules · 105
- search ellipse · 106
 - angle · 106
 - anisotropy · 107
 - radius 1 · 106
 - radius 2 · 106
 - X axis · 106
- search options
 - defining · 108
- search rules
 - blanking values · 106
- second derivative · 424
- select all · 44, 48, 476
- select axis · 33
- select block of cells · 62
- select case...end select · 518
- select entire worksheet button · 57
- select maps · 379
- select objects · 475–77
 - block select · 476
 - bounding box · 477
 - CTRL+click · 475, 477
 - delete · 492
 - deselect · 478
 - invert · 476
 - multiple · 475
 - object manager · 491
 - overlapping · 475
 - SHIFT key · 475
 - spacebar · 475
- select validation points within these limits · 102
- selected objects only · 463
- selecting
 - axes · 348
- selecting cells · 62
- selecting objects
 - aperture · 549
 - rectangle must fully surround · 549
- selection · 549
- selection collection · 504, 506
- Selection collection · 500
- selection handles · 477, 549
- separation angle · 166
- separation distance · 166, 185
- separator · 85
- serial number · 16
- server installation · 13
- set · 209, 214, 221, 227
- SET · 552, 559
- set file · 559
- set next statement · 539
- set statement · 516
- set values in blanked grids to · 456
- set values outside grid to · 455
- setting files · 552
 - custom · 559
- setting persistence · 553
- setting value · 552
- shaded relief map · 9, 267–72, 363
 - blanked regions · 267
 - central difference · 271
 - color spectrum · 363
 - colors · 271, 364
 - components · 268
 - creating · 269
 - dither bitmap · 272
 - general page · 269
 - gradient method · 271
 - grid file density · 267
 - horizontal light position · 270
 - input grid file · 269
 - Lambertian reflection · 271
 - light position angles · 270
 - light source · 270
 - Lommel-Seeliger law · 271
 - midpoint difference · 271
 - missing data · 272
 - overlays · 272
 - Peucker's approximation · 271

- properties · 269
- reference · 272
- reflectance · 267
- relief parameters · 271
- shading method · 271
- simple shading · 271
- USGS DEM · 268
- vertical light position · 270
- Z scale factor · 271
- shading method
 - shaded relief map · 271
- shape object · 499, 543
- shapefile · 232
- shapes · 469–74
- shapes collection · 506
- Shapes collection · 500
- shapes object · 496
- shear · 488, 489
- sheet · 511
 - close all · 511
 - open uses · 511
- Shepard's method · *See* modified Shepard's method
- shift and difference filters · 393
- shift cells · 63
- SHIFT key · 475
- SHIFT+F1 · 511
- shininess · 318
- show base · 297, 311
 - surface · 319, 325
- show color scale · 263, 264, 286, 328
 - surface · 319
- show contours · 406
 - grid node editor · 406
- show grid · 551
- show in window
 - statistics · 71
- show legend · 249
 - classed post map · 254
- show lower surface · 297
- show margins · 550
- show next statement · 539
- show nodes
 - grid node editor · 406, 407
- show overlays · 319
- show page rectangle · 550
- show position · 551
- show rulers · 551
- show scale legend
 - vector map · 290
- show upper surface · 297
- show vertical lines · 297, 311
- SHP · 232, 585, 592
- SHX · 592
- sill · 168, 169
- simple planar surface · 131
- simple shading method · 271
- Simpson's 3/8 rule · 445, 446, 448
- Simpson's rule · 445, 446, 447
- simulate mouse operations with keyboard · 487
- simulate stock hatch patterns · 547
- sine · 562
- single · 515
- single-use tool mode · 548
- size · 338, 468, 488
 - symbol · 557
 - symbol properties · 479
 - text · 556
 - text properties · 479
- skip · 209, 214, 221, 227
- skip grid rows and columns · 412
- slice · 443, 454, 455, 454–57
 - blanking file · 455
 - data file · 454
 - plotting · 456
- slider button · 365, 366
- SLK · 65, 594
- slope · 427, 428, 430, 434
 - default linear variogram · 193
 - linear variogram model · 185
 - variogram · 170
- smooth
 - spline · 383–87
- smooth contour lines · 229
- smooth lighting · 321

Index

- smoothing · 156, 199, 201
 - inverse distance to a power · 114, 115
- smoothing factor · 126
- smoothing grids · 97
- smoothing interpolators · 156, 157, 185
- smoothing variograms · 178
- snap to ruler · 375, 551
- solid handles · 477
- sort first by · 68
- sort last by · 68
- sort next by · 68
- sort worksheet · 67
- sorting rank · 67
- spacebar · 475, 487
- spacing · 95, 98, 99, 164, 400
 - contour labels · 223
 - ticks · 35
- spatial file formats · 466
- specular · 322
- spherical model · 184
- spherical variogram model · 180
- spline smooth · 383–87
 - expand grid · 385
 - recalculate grid · 386
- split lines · 514
- square · 472
- square root scaling
 - vector map · 284
- square root symbol scaling · 257
- SRF · 6, 36, 463, 466
- SrfDocPlot · 503
- SrfDocWks · 503
- stack · 506
- stack maps · 373, 375, 381
- stack window · 509, 540
- standard deviation · 76, 143, 564
- standard deviation filter · 394
- standard deviation of means · 73
- standard deviations
 - kriging · 119
- standard error of the mean · 73
- standardized variogram · 119, 176
- standardized variogram estimator · 177
- starting in cell · 71
- starting Surfer · 18
- statements · 514
- statistical functions · 563
- statistics · *See also* statistic name
 - cross validation · 102
 - references · 80
 - variogram · 173
 - worksheet · 71–80
- statistics page
 - variogram · 191
- statistics report · 95
- status bar · 24, 38, 475, 510, 536
- step amount · 175
- step CCW · 175
- step CW · 175
- step into · 538
- step out · 539
- step over · 539
- step to cursor · 539
- store residuals · 458
- stretching and scale · 338
- strikethrough · 570
- string · 515
- STX · 160
- style · 468, 555
 - font · 556
 - line properties · 479
 - numeric format · 558
 - text properties · 479
 - vector map · 282
- sub main · 509
- subroutines · 520, 522, 539
 - writing · 521
- subscript · 468, 571
 - example · 573
- subset · *See* limits
- subset of grid · 411
- subtitle
 - variogram model · 192
- subtraction · 561
- suffix · 359, 558
 - label format · 480

- suggested reading · 195, 535
- sum · 72, 93, 144, 564
- superscript · 468, 571
 - example · 573
- support forum · 15
- surface · 10, 315–26, 363, 365
 - add color scale bar · 328
 - ambient light · 322
 - base · 316, 325
 - base properties · 325
 - base stitching · 327
 - blanked nodes · 318
 - change color · 50
 - color · 316
 - color modulation · 324
 - color scale bar · 328–29
 - color scale bar visibility · 328
 - creating · 47, 316
 - diffuse light · 322
 - edit color scale bar · 329
 - flat lighting · 321
 - general page · 317
 - horizontal light position · 322
 - light color · 322
 - light position angles · 321
 - light source · 321
 - linear resampling · 324
 - lower color · 318
 - material · 318
 - material color · 50
 - memory · 327
 - mesh lines · 49, 319
 - mesh surface offset · 320
 - mesh tips · 320
 - nearest resampling · 324
 - overlay resolution · 324
 - overlay visibility · 324
 - overlays · 47, 315, 323, 327
 - printing · 327
 - properties · 47–51, 317–26
 - remove color scale bar · 328
 - resample · 324
 - shininess · 318
 - show base · 319
 - show color scale · 319, 328
 - show overlays · 319
 - smooth lighting · 321
 - specular light · 322
 - upper color · 50, 318
 - vertical light position · 322
- surface area · 443, 445, 449, 448–50
 - negative · 449
 - positive · 449
- surface components · 316
- surface definition · 131
- surface map fill · 363
- surface offset · 320
- Surfer 7
 - grid file format · 579–81
- Surfer automation help · 495, 499, 511
- Surfer files · 6, 36, 466
 - combine · 375
- Surfer grids · 575–81
- Surfer objects · 499, 502
 - using · 523
- surfer.set · 552
- SYLK · 65, 594
- symbol · 235, 247, 471, 572
 - classed post map · 253
 - color · 557
 - size · 557
 - variogram model · 192
- symbol angle · 245, 249
 - post map · 247
- symbol freq
 - classed post map · 249
- symbol height
 - post map · 257
- symbol index · 244, 245
- symbol locations · 243
- symbol number · 245
- symbol origin · 284
- symbol page
 - vector map · 282
- symbol properties · 43, 242, 479
 - base map · 233

Index

- default · 550
- default settings · 557
- symbol set · 244, 557
- symbol size · 43, 248, 256
 - legend · 256
- symbols from data file · 244
- Symphony · 594
- syntax
 - math text instructions · 569
- system requirements · 11

T

- tab delimited · 82
- tagged image · 232
- tagged image format (TIFF) · 589
- tangent · 562
- tangential curvature · 432
- Targa (TrueVision) · 232, 589
- target precision
 - variogram autofit · 189
- technical support · 15
- telephone · 16
- template
 - legend · 255
- terrain aspect · 146, 429
- terrain model · 427–33
 - plan curvature · 431
 - profile curvature · 430
 - references · 433
 - tangential curvature · 432
 - terrain aspect · 429
 - terrain slope · 428
- terrain slope · 146
- terrain statistics · 146
- text · 55, 235, 467–68, 514
 - clipboard format · 462
 - color · 571
 - equations · 573
 - Greek characters · 468
 - math text instructions · 569
 - properties · 479
 - size · 556
 - subscript · 468, 571
 - superscript · 468, 571
- text blocks · 585
- text ID · *See* object ID
- text properties
 - default · 550
 - default settings · 556
- text qualifiers · 82
- text strings · 514
- texture maps · 323
- TGA · 232, 585, 589
- Thiessen polygons · 127
- thin out grid · 412
- thin plate spline · 132, 134
- third quartile · 73
- thousands · 359
- thousands separator · 66
 - label format · 480
- threshold · 427
- threshold averaging filter · 395
- tick
 - labels · 35
 - spacing · 35
- tick labels · 357
- tick spacing · 350
- ticks · 353
- TIF · 232, 466, 585, 589
- tilt · 38, 200, 242, 260, 268, 274, 295, 316, 331, 332, 333
- time · 468, 572
- tips on selecting cells · 62
- title · 34
 - axis · 352
 - dialog · 46
 - legend · 254
 - vector map legend · 290
- to keep · 93
- toggle break · 539
- tolerance
 - variogram · 175
- toolbars · 5
- top axis · 347
- topological vector profile · 160

total page count · 85
total planar area · 445
trace function · 540
trackball · 38, 332, 335
transform · 69, 412–15, 483, 488
 clear transforms · 489
 examples · 70, 565
 identity · 489
 mirror X · 415
 mirror Y · 415
 offset · 413
 order · 489
 references · 489
 rotate · 414
 rotation · 489
 scale · 413
 shear · 489
 X offset · 488
 X scale · 488
 Y offset · 488
 Y scale · 489
transform data · 561
transform dialog · 564
transformations · 488
transparent · 213
trapezoidal rule · 445, 446, 447
treat text and empty cells as 0.0 · 69
triangulation · 127, 135
triangulation with linear interpolation · 135–
 37, 151, 155, 156
 references · 137
 setting options · 136
trigonometric functions · 562
trim · *See* limits or blanking
troubleshooting installations · 14
true color · 464
tutorial · 17–51
two grid vector maps · *See* vector map
TXT · 65, 82, 593
type library
 adding · 532
 defined · 532
type method · 500

type statement · 517
typeface · 468, 570
 text properties · 479

U

U.S. Geological Survey · *See* USGS
unblank node
 grid node editor · 408
underline · 570
undo · 548
undo levels · 548
uniform line properties · 209
uninstalling Surfer · 14
United States Geological Survey · 160
units
 page · 550
unselect · *See* deselect
update a post or classed post map · 253
updating Surfer · 14
upper color · 50
 surface · 318
upper quartile · 142
upper quartile filter · 393
upper surface · 297, 443, 445, 446
use a color spectrum file · 370
use data limits · 258, 366
 map limits · 344
use only spatial information · 466
user defined
 binning method · 252
user defined polynomial · 131
user dialog box · *See* dialog box
user input · 524
user-defined filters · 391
user-defined type · 517
UserDialog editor · 528
USGS · 160
USGS DEM · 89, 160, 259, 268
USGS digital line graph · 232, 588
USGS topological vector profile · 588
using Surfer · 18
using the keyboard · 487

Index

using variogram results in kriging · 194

V

value · 365

variable · 515

 change value · 538

 view value · 538

variable type · 516

variables · 515–17

variance · 75, 143

 variogram · 170

variance filter · 394

variance line

 variogram model · 192

variant · 515

VarioComponent object · 543

variogram · 168, 176, 163–95, 561

 angular divisions · 173

 autocorrelation · 119

 Boolean expressions · 94

 creating · 171

 data page · 172

 detrend · 174

 direction · 168

 experimental curve · 170

 exporting · 194

 general page · 173

 histogram · 191

 length or slope · 170

 linear detrending · 174

 max lag distance · 173

 new variogram · 172–74

 nugget effect · 168, 169

 observations · 176

 pair order · 176

 pairs · 170

 properties · 171

 quadratic detrending · 174

 radial divisions · 173

 range · 168

 references · 195

 report · 191. *See* online help

 scale · 169

 separation distance · 168

 sill · 168, 169

 smoothing · 178

 standardized variogram · 119

 statistics · 173

 value · 168

 variance · 170

variogram autofit

 default method · 189

 fit criteria · 189

 initial · 190

 least absolute value · 189

 least squares · 189

 max lag distance · 190

 maximum · 190

 maximum distance · 190

 maximum iterations · 189

 minimum · 190

 parameter fit limits · 190

 restarting · 189

 target precision · 189

variogram axis · 347

variogram components · 117

variogram functions · 180

variogram grid · 165, 173

 example · 167

 memory · 165

 separation angle · 166

 separation distance · 166

variogram model · 117

 cubic · 180

 defined · 169

 error variance · 186

 exponential · 180

 Gaussian · 180

 graphical example · 169

 line · 192

 linear · 180, 185

 logarithmic · 180

 micro variance · 186

 model line · 192

 nesting · 180

- nugget effect · 180, 185
- pairs · 192
- pentaspherical · 180
- power · 180
- quadratic · 180, 184
- rational quadratic · 180
- spherical · 180, 184
- subtitle · 192
- symbols · 192
- variance line · 192
- wave (hole effect) · 180
- variogram modeling
 - overview · 163
- variogram models
 - scale · 184
- variogram object · 543
- variogram properties · 175–92
 - anisotropy · 186
 - autofit model · 188–90
 - estimators · 176
 - experimental page · 175–80
 - lag direction · 175
 - lag width · 178
 - max lag dist · 178, 179
 - model page · 180–87
 - number of lags · 178, 179
 - plot page · 192
 - separation distance · 185
 - statistics page · 191
 - step amount · 175
 - step direction · 175
 - tolerance · 175
 - vertical scale · 180
- variograms · 118
- VBA compatibility · 534
- vector colors · 283
- vector example · 276
- vector map · 9, 363
 - 1-grid · 273
 - 2-grid · 273
 - 2-grid data · 275–78
 - add color scale bar · 286
 - angle · 282
 - angle (direction) data · 277
 - arrow direction · 274
 - arrow length · 274
 - axes · 289
 - Cartesian coordinates · 281
 - Cartesian data · 275
 - clip symbols · 282
 - clip symbols at map limits · 288
 - color · 274
 - color scale bar · 274, 286–87
 - color scale bar visibility · 287
 - color scaling · 283
 - colors · 283, 365
 - components · 274
 - creating · 279
 - data page · 280, 281
 - edit color scale bar · 286
 - edit legend · 290
 - fill symbols · 363
 - legend · 274, 285, 290–91
 - legend frame · 291
 - legend title · 290
 - length (magnitude) data · 277
 - limits · 288
 - linear scaling · 284
 - logarithmic · 284
 - magnitude · 285
 - negative values and polar grids · 277
 - polar coordinates · 281
 - polar data · 277
 - properties · 279–85
 - reference vectors · 291
 - remove color scale bar · 287
 - reverse vectors · 284
 - scaling · 284
 - scaling method · 284
 - show color scale · 286
 - square root · 284
 - style · 282
 - symbol frequency · 283
 - symbol origin · 284
 - symbol page · 282, 288
 - X and Y component data · 275

Index

- version number · 16
- vertical light position
 - shaded relief map · 270
 - surface · 322
- vertical scale · 180
- view · 38, 331, 334, 332–36, 373, 381
 - always split · 509
 - colors · 509
 - examples · 336
 - fit to window · 25
 - object manager · 42, 475, 490
 - status bar · 510, 536
 - zoom selected · 34
- view data · 95
- view errors · 536
- view map coordinates · 239
- visibility
 - axis · 349
 - overlay · 380
- visible property · 495, 502
- visible status · 490
- volume · 436, 443–53
 - between two surfaces · 451
 - blanked regions · 446, 448, 453
 - cut · 445
 - cut and fill · 446
 - cut minus fill · 445
 - extended Simpson's 3/8 rule · 448
 - extended Simpson's rule · 447
 - extended trapezoidal rule · 447
 - fill · 445
 - improving accuracy · 453
 - increasing accuracy · 444
 - negative · 445, 446
 - net · 446
 - of a lake · 452
 - positive · 445, 446
 - prisms · 444
 - reference · 449
 - Simpson's 3/8 rule · 445
 - Simpson's Rule · 445
 - trapezoidal rule · 445
 - units · 448

- upper and lower surface · 445
- with blanked grids · 453
- Z scale factor · 450

W

- watch window · 509, 538
- wave (hole-effect) variogram model · 180
- wave number · 441
- web · 16
- weighted average interpolation · 157
- weighted averaging · 156
- weighting
 - anisotropy · 108
 - inverse distance to a power · 114, 115
- weighting neighbors · 126, 127
- when grid is completed · 548
- while...wend · 518
- width · 555
 - line properties · 479
- window
 - object · 499
- window object · 505
- window state · 503
- window types · 4
- Windows bitmap · 232, 589
- Windows clipboard (picture) · 232
- Windows collection · 500, 503, 504
- Windows metafile (picture) · 232
- Windows metafiles · 589
- windows property · 503
- wireframe · 10, 293–313
 - adding, deleting, modifying Z levels · 305
 - apply zones to lines of constant · 300
 - base · 294, 297, 311
 - blanked regions · 313
 - border · 298
 - color scale bar · 301, 309–10
 - color zones · 40, 294, 306–9
 - color zones page · 300
 - components · 294
 - contour levels · 307
 - creating · 37

- elevation · 311
 - flat areas · 313
 - general page · 296
 - grid file · 294
 - hidden lines · 378
 - level file · 299, 301
 - levels · 299, 300, 304
 - limits · 293, 342
 - line properties · 40, 299, 300, 308
 - line property precedence · 303
 - line props · 311
 - lines · 301
 - overlying · 373
 - plot lines of constant · 296, 301, 302
 - properties · 37–41, 295, 296–301
 - remove hidden lines · 298
 - rotation · 38
 - scale · 39
 - show base · 311
 - show lower surface · 297
 - show upper surface · 297
 - show vertical lines · 297, 311
 - smoothing · 312
 - tilt · 38
 - view · 38
 - X lines · 294
 - Y lines · 294
 - Z levels page · 298
 - Z line properties · 303–6
 - Z lines · 294
 - wireframe map
 - object · 496
 - wireframes
 - overlays · 313
 - WksDocument object · 505
 - WksWindow · 500
 - WMF · 232, 466, 589
 - WordPerfect bitmap · 232, 589
 - working with data · 58
 - worksheet · 56, 53–88
 - active cell · 58
 - alignment · 67
 - calculations · 69
 - center on page · 84
 - create data file · 21
 - data format · 53
 - enter data · 58
 - error codes · 566
 - gridlines · 85
 - headers and footers · 84
 - keyboard commands · 58–60
 - limits · 88
 - margins · 84
 - memory requirements · 88
 - modify · 64
 - numeric formatting · 66
 - opening data into · 61
 - options · 85
 - page setup · 83–86
 - printing · 83
 - selecting cells · 62
 - size · 88
 - sort · 67
 - worksheet column for labels · 46, 250
 - worksheet columns · 247, 249
 - worksheet equations · 564
 - worksheet formatting
 - save · 64
 - worksheet labels
 - math text · 569
 - worksheet name · 57
 - worksheet statistics · 71–80
 - worksheet window · 4
 - components · 57
 - worksheet windows · 56
 - WPG · 232, 585, 589
 - writing functions · 522
 - writing scripts · 512
 - writing subroutines · 521
-
- X**
- X axis · 347
 - search ellipse · 106
 - X component data · 275
 - X component grid · 275

Index

X coord · 247, 249
X coordinate · 454
X direction · 95, 400
X grid lines · 89
X lines · 293, 294
X offset · 413, 488
X scale · 337, 413, 488
X tolerance
 duplicate data · 94
X, Y, Z histogram
 variograms · 191
XLS · 65, 81, 594
xor · 94, 564
XY coordinates · 243
X-Y plane · 250
XYZ data · 6, 7
 to grids · 89
XYZ data file · 19, 95
XYZ data files · 6, 53, 96
 creating grids from · 91

Y

Y axis · 347
Y component data · 275
Y component grid · 275
Y coord · 247, 249
Y coordinate · 454
Y direction · 95, 400

Y grid lines · 89
Y lines · 293, 294
Y offset · 413, 488
Y scale · 337, 413, 489
Y tolerance
 duplicate data · 94

Z

Z coordinate
 missing · 54
Z level · 305, 309
Z level dialog · 206
Z levels · 298, 299, 300, 305, 371
Z line properties · 303–6
Z lines · 293, 294
Z moment statistics · 143
Z order statistics · 142
Z scale · 39, 337
Z scale factor · 450
 shaded relief map · 271
Z value · 454
 image map · 259
Z value array · 158
Z value of grid at a point · 459
Z values · 6, 244
 grid node editor · 4
zoom selected · 34